

UNIVERSIDADE FEDERAL DO PARANÁ

FRANCISCO BARTOSIEWICZ NETTO

OTIMIZAÇÃO DO TEMPO TOTAL DE TRABALHO NA
PROGRAMAÇÃO DA PRODUÇÃO EM OFICINA DE MÁQUINAS COM
TEMPO DE PREPARAÇÃO DEPENDENTE DA SEQUÊNCIA

CURITIBA

2014

FRANCISCO BARTOSIEVICZ NETTO

**OTIMIZAÇÃO DO TEMPO TOTAL DE TRABALHO NA
PROGRAMAÇÃO DA PRODUÇÃO EM OFICINA DE MÁQUINAS COM
TEMPO DE PREPARAÇÃO DEPENDENTE DA SEQUÊNCIA**

Tese apresentada ao Programa de Pós-Graduação em Métodos Numéricos em Engenharia, Área de Concentração em Programação Matemática, Setores de Tecnologia e de Ciências Exatas da Universidade Federal do Paraná, como requisito parcial à obtenção do grau de Doutor em Ciências.

Orientador: Prof. Dr. Arinei Carlos Lindbeck Silva

CURITIBA

2014

TERMO DE APROVAÇÃO

FRANCISCO BARTOSIEVICZ NETTO

OTIMIZAÇÃO DO TEMPO TOTAL DE TRABALHO NA PROGRAMAÇÃO DA PRODUÇÃO EM OFICINA DE MÁQUINAS COM TEMPO DE PREPARAÇÃO DEPENDENTE DA SEQUÊNCIA

Tese aprovada como requisito parcial para obtenção do grau de Doutor no Programa de Pós-Graduação em Métodos Numéricos em Engenharia - Área de Concentração em Programação Matemática, Setores de Tecnologia e de Ciências Exatas da Universidade Federal do Paraná, pela seguinte banca examinadora:



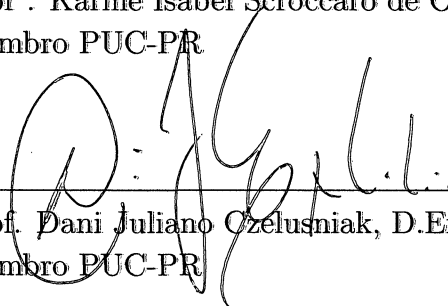
Orientador: Prof. Arinei Carlos Lindbeck da Silva, D. Eng.
Membro UFPR



Prof. Ademir Aparecido Constantino, D.Eng.
Membro UEM



Prof^a. Karine Isabel Scroccaro de Oliveira, D.Eng.
Membro PUC-PR



Prof. Dani Juliano Ozelusniak, D.Eng.
Membro PUC-PR



Prof^a. Deise Maria Bertholdi Costa, D.Eng.
Membro UFPR

Curitiba, 10 de Dezembro de 2014.

Dedicatória

Ao meu pai, Silvio Bartosiewicz.

In Memoriam.

Agradecimentos

Agradeço antes de tudo e todos ao meu Deus, por minha esposa, família e amigos, e que desde sempre me capacitou e me honrou para chegar neste momento. Obrigado Senhor por este momento tão especial em minha vida. A Ti Senhor, toda a honra!

À minha mãe Regina, que sempre se sacrificou para me criar da melhor forma possível. Ao meu pai Silvio (em memória), que sempre me ensinou a ser um homem melhor e honrado.

Agradeço minha amada esposa Rose, que soube me suportar quando quis esmorecer, mas que acima de tudo, me amou quando eu mais precisei. Sem você este dia jamais chegaria. A você, todo meu amor e carinho.

Ao meu amigo e orientador Arinei que soube ser compreensivo quando enfrentei toda luta pelo meu pai, e enérgico quando deveria retomar meu caminho e terminar esta jornada. Aos professores que gentilmente aceitaram avaliar meu trabalho: Ademir, obrigado pelo seu rigor matemático, tão importante e que muitos não dão o devido valor; Karine, pelo incentivo e força para eu não desistir; Dani Juliano, pela disposição em sempre me ajudar; Deise, pelo carinho e ensinamentos de tantos anos para com todos os seus alunos; e Fabiano, pelos elogios tão especiais sobre a qualidade do meu trabalho. A vocês meus tutores, minha eterna gratidão.

Não poderia deixar de agradecer a Maristela e ao Jair, que sempre me ajudaram com tanta disposição na secretaria do curso em todos estes anos (muitos anos).

Ainda agradeço a meus melhores amigos Thiago, Danilo, Pedro e Jonathan, que sempre me animaram já que a *zoeira never ends*. A vocês meu *huehuehue brbr*.

Francisco Bartosiewicz Netto

Epígrafe

“Se você confessar com a sua boca que Jesus é Senhor e crer em seu coração que Deus o ressuscitou dentre os mortos, será salvo.”.
(Romanos 10:9)

Resumo

O planejamento da produção de uma indústria é ponto fundamental na busca de bons resultados para metas estabelecidas e maximizar lucros. Para tal, o Planejamento e Controle da Produção é uma área de estudo que pode, com auxílio da Pesquisa Operacional e seus modelos matemáticos, otimizar diversos objetivos. Dentre as diversas decisões a serem tomadas dentro da indústria, uma é crucial para o sucesso ou não: a programação da produção, através da determinação de como sequenciar as tarefas pelas máquinas que as mesmas devem passar para então chegar ao produto final. São diversos ambientes descritos na literatura, sendo o mais abrangente a oficina de máquinas. E quando neste ambiente, considera-se um possível tempo de preparação da máquina antes de receber a próxima tarefa, e este tempo depende da última tarefa sequenciada nesta máquina, o problema torna-se ainda maior e mais complexo. Para resolver este problema, este trabalho tem como objetivo apresentar uma nova heurística que busca minimizar o tempo total de trabalho na situação descrita. Esta heurística é implementada e testada para problemas de diversos tamanhos e os resultados obtidos são satisfatórios.

Palavras-chave: Programação da Produção, Oficina de máquinas, Tempo de preparação, Dependência da sequência.

Abstract

An industry's planning of production is a fundamental area when seeking the best results for reaching established goals and maximizing profits. For that, with the help of Operational Research and its mathematical models, the planning and controlling of production is an area of study that can optimize many different areas. Amongst all these decisions made in the industry, one of them is quintessential in determining whether one succeeds or fails inside the industrial market - the scheduling of the production, through deciding how to sequence the tasks of the machines through which the product undergoes until it reaches its final state, completed and assembled. For this, there are many different environments in which machine sequencing is needed, but the most encompassing of them is the job shop. In this type of environment, each machine needs some sort of preparation prior to performing a certain task, and this preparation is dependant on the task that was previously performed by that machine - a matter which further complicates deciding the right sequencing. To solve this problem, this study presents a new heuristic that seeks to minimize the total time required for a complete a task similar to the one described earlier - one that considers a set of tasks with preparation dependant on the previous task performed. This heuristic is implemented and tested on several different problems of varying size, all of which show very succesful results.

Key-words: Scheduling, Job shop, Setup time, Sequence dependence.

Lista de Figuras

2.1	Configuração dos ambientes	10
3.1	Cenários para a Etapa 3	26
3.2	Cenários 1 e 2 para a Etapa 3	27
3.3	Cenários 3 e 4 para a Etapa 3	27
3.4	Cenário 3 para a Etapa 3	28
3.5	Cenário 5 para a Etapa 3	28
3.6	Interface com o usuário <i>FBN_JobShopScheduling_Tese</i>	30
3.7	Interface de apresentação de resultados	33
4.1	Comparativos dos resultados 3x3 a 7x7	40
4.2	Resultados 8x8	41
4.3	Resultados 9x9	42
4.4	Resultados 10x10	44
4.5	Resultados 15x15	44
4.6	Resultados 20x20	45

Lista de Tabelas

2.1	Resumo da literatura sobre a programação da produção	18
4.1	Tempos médios de processamento (ms) 3x3 a 7x7	38
4.2	Intervalos de Confiança - tempos médios de processamento (ms) 3x3 a 7x7	39
4.3	Comparativos resultados 3x3 a 7x7	39
4.4	Comparativos resultados 10x10, 15x15 e 20x20	43
4.5	Tempos médios das Iterações (ms) 3x3 a 20x20	46
B.1	Resultados Testes 3x3	57
B.2	Resultados Testes 4x4	59
B.3	Resultados Testes 5x5	60
B.4	Resultados Testes 6x6	61
B.5	Resultados Testes 7x7	62
B.6	Resultados Testes 8x8	63
B.7	Resultados Testes 9x9	64
B.8	Resultados Testes 10x10	65
B.9	Resultados Testes 15x15	66

B.10 Resultados Testes 20x20	67
--	----

Lista de Siglas

AG - Algoritmo Genético

BT - Busca Tabu

B&B - *Branch and Bound*

B&C - *Branch and Cut*

EDD - *Earliest Due Date*

GAMS - *General Algebraic Modeling System*

GBD - *Generalized Benders Decomposition*

GRASP - *Greedy Randomized Adaptive Search Procedure*

HE - Heurística

MHE - Meta-heurística

OCF - Otimização por Colônia de Formigas

ONP - Otimização por Nuvem de Partículas

PIM - Programação Inteira Mista

PCP - Planejamento e Controle da Produção

SA - *Simulated Annealing*

SM - Simulação

SPT - *Shortest Processing Time*

VNS - *Variable Neighborhood Search*

WSPT - *Weighted Shortest Processing Time*

Sumário

1	Introdução	1
1.1	O problema	1
1.2	Objetivos	3
1.2.1	Objetivo geral	3
1.2.2	Objetivos específicos	3
1.3	Justificativa e Contribuição do Tema	4
1.4	Metodologia da Pesquisa	5
1.5	Estrutura do trabalho	5
2	Revisão Bibliográfica	7
2.1	Programação da Produção	7
2.1.1	Decisões	8
2.1.2	Ambientes	8
2.1.3	Objetivos da Programação	11
2.1.4	Regras de Prioridade	12
2.2	Revisão da literatura	13

2.3	Algoritmo de Giffler e Thompson	20
3	Metodologia	23
3.1	O Algoritmo	23
3.2	A Implementação	29
3.3	O Modelo de Programação Matemática	34
4	Resultados	37
4.1	Problemas 3x3 a 7x7	37
4.2	Problemas 8x8 e 9x9	39
4.3	Problemas 10x10, 15x15 e 20x20	42
5	Conclusões	47
	Referências Bibliográficas	49
A	Modelo Matemático	53
B	Resumo dos resultados	57

Introdução

O trabalho desta tese encontra-se dentro da classe de problemas de sequenciamento — *scheduling* — da produção, e trata de uma variação do problema descrito na literatura, porém, com poucos métodos de resolução efetivos e de fácil implementação, algo necessário para aplicações reais.

1.1 O problema

A produção de bens de consumo, como a conhecemos hoje, somente teve início com a Revolução Industrial, quando foi possível produzir e criar meios para o consumo em massa. Os sistemas de Planejamento e Controle da Produção (PCP) evoluíram como fruto da evolução da própria ciência da administração, desde os esforços de Frederick W. Taylor e Henri Fayol, no início do século XX.

Dos trabalhos produzidos por Taylor, surgiram diversos trabalhos subsequentes, como o de Henry Gantt, que desenvolveu um sistema de programação de produção baseado em gráficos e cálculos, muito utilizado até os dias de hoje através do famoso Gráfico de Gantt.

Ainda segundo Lustosa *et al.* (2008), o PCP têm três grandes funções: o planejamento, a programação e o controle da produção. Sendo que na programação da produção que encontram-se os problemas de sequenciamento da produção.

O problema de sequenciamento pode ser entendido, num primeiro momento, como o procedimento que utiliza-se de regras de decisão lógica para selecionar uma ordem de produção, ou seja, que tarefa (objeto, produto, etc.), dentre as disponíveis para ser processada, será alocada a uma máquina que esteja livre para processar e em que momento este processamento deve ocorrer.

Conforme descrito por Colin (2007), as regras que estabelecem a forma que o sequenciamento deve ocorrer, pertencem aos primórdios da administração produtiva e ainda hoje podem ser consideradas um poderoso instrumental a ser usado no dia a dia. A teoria em si foi iniciada na primeira metade da década de 1950, com três trabalhos publicados entre 1954 e 1956 que influenciaram de maneira profunda todo o desenvolvimento teórico posterior. Nestes trabalhos, regras para a prioridade do sequenciamento, como *Shortest Processing Time* (SPT), *Weighted Shortest Processing Time* (WSPT) e *Earliest Due Date* (EDD), foram definidas e estudadas formalmente. As regras já eram conhecidas nos ambientes profissionais, mas até então não haviam sido estudadas formalmente.

Proud (1999), descreve que o problema de sequenciamento da produção tomou forma e começou a ser estudado de fato na década de 1960, após uma nova geração começar a se formar no pós Segunda Grande Guerra Mundial. As companhias norte-americanas começaram a se preocupar com o uso otimizado de insumos na produção de seus produtos, com o objetivo de manufaturar apenas a quantidade necessária para a venda dos mesmos. Infelizmente, a manufatura raramente produzia apenas o que seria vendido. E com a invariável mudança de demanda, a quantidade dos insumos também mudava junto. Com o desenvolvimento de ferramentas computacionais, tornou-se possível gerir a esmagadora mudança de planejamento que as plantas de produção e os fornecedores até então não podiam lidar.

Problemas de programação¹ e sequenciamento surgem em qualquer ambiente em que se deseja definir horários, locais e atividades a serem realizadas, como por exemplo:

- Sequenciamento da chegada de aviões a aeroportos ou de navios a portos.
- Sequenciamento dos pedidos dos clientes no chão de fábrica (qualquer empresa de produção discreta como automobilística, de eletroeletrônicos, metalomecânica, etc.).
- Sequenciamento das atividades a serem executadas por um *software*.
- Sequenciamento dos clientes que estão esperando na fila.
- etc.

¹Programação está associada ao estabelecimento de horários de início e término das atividades

Muitas vezes, o sequenciamento exige uma preparação prévia para poder então ser aplicada a execução da atividade, o que torna-se um fator complicador do problema de sequenciamento. Quando este tempo de preparação varia conforme muda-se a sequência, tem-se uma complicação a mais. Essa situação é frequente, por exemplo, em uma linha de produção de massas. Quando se passa uma massa de pão por uma batedeira industrial, esta é facilmente retirada e limpa para a próxima massa ser batida. Quando se passa uma massa de panetone pela batedeira, esta necessita de uma limpeza especial, pois ficam várias partes da massa aderida na batedeira.

Esse tempo de preparação da máquina, em função da sequência, é chamado de tempo de *setup*, que conforme a sequência assumida pode ter seu tempo variável. É esta classe de problema considerada por esta proposta de tese.

1.2 Objetivos

1.2.1 Objetivo geral

Este trabalho tem como objetivo principal desenvolver um algoritmo heurístico, baseado em Giffler e Thompson (1960), para otimizar o problema de sequenciamento em oficina de máquinas com tempo de *setup* variável.

1.2.2 Objetivos específicos

Para alcançar o objetivo geral apresentado, sugerem-se os seguintes objetivos específicos:

- Efetuar a revisão da literatura sobre o assunto de sequenciamento de tarefas.
- Propor um algoritmo heurístico para a resolução do problema.
- Implementar o algoritmo proposto, visando minimização do tempo de processamento.

- Aplicar o algoritmo implementado, mostrando sua eficiência em resolver o problema, mesmo para casos grandes.
- Utilizar a solução obtida pela heurística como ponto factível inicial para o modelo exato buscar o ótimo do problema, gerando um modelo híbrido: heurístico-exato.

1.3 Justificativa e Contribuição do Tema

O problema de sequenciamento de tarefas em máquinas com ordens variáveis, considerando os tempos de preparação/ajustes destas máquinas, conforme a tarefa anteriormente executada na máquina com relação à que entrará na mesma é de importância crucial no dias de hoje nos ambientes produtivos.

A busca pela otimização de recursos, através da minimização do tempo ocioso das máquinas em um planejamento de produção, impacta diretamente na alocação de mão-de-obra, nas quantidades a serem estocadas, nos gargalos que a produção pode apresentar, dentre outros. Sendo assim, entende-se que este trabalho, com o algoritmo apresentado, justifica a sua necessidade e contribuição, não só para a ciência como um todo, por avançar um pouco mais a busca de soluções eficientes e rápidas, mas também por ser de cunho totalmente aplicável dentro das empresas e indústrias.

A pesquisa em artigos e publicações foi muito além da apresentada nesta proposta de tese. Os artigos citados foram os considerados mais relevantes em cada área que se propuseram a trazer avanço científico e tem a intenção de apresentar quão grande é o escopo de problemas e seus subproblemas dentro da Programação da Produção.

Porém, em momento algum desta vasta pesquisa foi encontrada uma heurística independente (ou não) de parâmetros subjetivos (dependentes da escolha do programador) que, de forma rápida e simples, otimize o problema de minimizar o tempo total de trabalho da programação da produção no ambiente *job shop* levando em conta o tempo de preparação dependente da sequência. Visto que não foi encontrado na literatura pesquisada algoritmo que trabalhe em cima do tema proposto, entende-se que há o ineditismo e a relevância necessários no trabalho.

1.4 Metodologia da Pesquisa

Inicialmente, fora estudado o algoritmo apresentado por Giffler e Thompson (1960), onde esse foi implementado e testado, com o objetivo de melhorar a compreensão do ambiente do problema de pesquisa. A partir desta primeira implementação, foi proposto o algoritmo apresentado nesta tese. Após esta etapa, buscou-se na literatura e periódicos científicos se houve ocorrência na comunidade científica algo semelhante à ideia desenvolvida. Nada foi encontrado, então iniciou-se a busca por publicações recentes sobre o assunto.

Foram realizadas buscas em periódicos e jornais como a *EJOR - European Journal of Operational Research* da *Elsevier* e o *Journal of Scheduling* da *Springer* com termos como “*job shop, scheduling, setup time, sequence dependence*”. Foram tratados cerca de 300 artigos, na busca de trabalhos na área e que eventualmente trouxessem algo próximo ao algoritmo elaborado.

Dos trabalhos mais recentes escolhidos, após a sua leitura e pesquisa também em suas referências bibliográficas, foram elencados os trabalhos que formam o corpo teórico, que sustenta o estudo de caso desenvolvido neste trabalho.

Com a pesquisa bibliográfica foi possível confirmar o ineditismo deste trabalho. Após a coleta de dados e o desenvolvimento da estrutura teórica, foram desenvolvidos os arranjos que culminam nas etapas de implementação e testes do algoritmo.

1.5 Estrutura do trabalho

O Capítulo 1 contém a introdução do trabalho. O Capítulo 2 apresenta uma descrição um pouco mais aprofundada dos elementos que cercam o problema e a revisão bibliográfica do assunto na literatura, na qual são apresentados os diversos tipos de problemas de sequenciamento existentes assim como métodos de resolução.

O Capítulo 3 apresenta como metodologia o algoritmo proposto para solucionar o problema tratado, sua devida implementação e o modelo exato considerado para validar

os resultados obtidos pela heurística. O Capítulo 4 traz todos os testes e resultados sobre os mesmos e as análises dos resultados. No Capítulo 5 as conclusões e sugestões para trabalhos futuros. A seguir, a bibliografia e referências utilizadas e os apêndices.

Revisão Bibliográfica

Este capítulo está dividido em duas seções, a primeira trata de uma breve revisão bibliográfica do problema da programação da produção dentro do PCP. Já a segunda, trata da revisão de publicações científicas de como este problema vem sendo tratado/resolvido dentro da comunidade científica ao longo dos últimos anos.

2.1 Programação da Produção

Elaborar a programação da produção pode ser definido como decidir quando e como cada tarefa deverá ser realizada para que se cumpram certos objetivos: reduzir estoques, entregas serem feitas no tempo certo, produção ser realizada no menor tempo possível, ou ainda, minimizar a ociosidade dos recursos na execução das tarefas de produção, podendo ser, por exemplo, máquina parada esperando para processar uma tarefa. Normalmente, os custos de não cumprir algum destes objetivos não é simples de se mensurar, o objetivo (na maioria das vezes) não é avaliado em termos de custo, mas por outros critérios operacionais, como minimizar atrasos, reduzir estoque e minimizar ociosidades.

A programação da produção é tida como uma operação de chão de fábrica, ou seja, faz parte das tarefas de execução produtiva na indústria. A sua abrangência é limitada no sentido de tempo, contemplando de horas a poucas semanas. Mas isso não a torna desprezível, afinal, uma empresa/indústria que não planeja, prevê e cumpre seus prazos de entrega certamente será esquecida em um mercado cada vez mais competitivo.

Este assunto é muito amplo, podendo contemplar a área de produção de bens de consumo, bem como prestação de serviços. O fato é que necessita-se criar modelos de apoio às tomadas de decisão que um gerente de produção precisa tomar dentro das soluções de programação para cada situação específica em que ele se encontrar. A Teoria

da Programação (*Scheduling*) trata dos problemas de otimização combinatória oriundos à programação da produção, bem como programação de entregas (roteirização de veículos, por exemplo), programação de escalas de trabalho, designação de tarefas, dentre outros.

2.1.1 Decisões

É possível categorizar os tipos de decisão dentro da programação de tarefas da produção, conforme Lustosa *et al.* (2008):

1. Designação (*assignment*): Determinação de onde a tarefa será executada.
2. Sequenciamento (*sequencing*): Determinação da sequência (ou ordem) em que as tarefas devem ser executadas em um dado recurso.
3. Programação (propriamente dita) (*scheduling*): Determinação de quando a tarefa deverá ser iniciada e terminada, não apenas a sequência.
4. Despacho (*dispatching*): Quando e para quem a ordem deverá ser emitida ou liberada.
5. Controle (*control*): Acompanhamento dos trabalhos para garantir que a programação se mantenha válida e se não se faz necessário ajustes ou reprogramação da produção.
6. Apressamento (*expediting*): Acelerar a ordem de produção pelo aumento de sua prioridade para manter a programação válida ou atender uma necessidade específica.
7. Carregamento de oficinas (*shop loading*): Definição dos roteiros e programação das tarefas dentro de uma fábrica, ou seção de um sistema produtivos.

2.1.2 Ambientes

Após essa categorização, faz-se necessário também categorização de que tipo de ambiente será a execução das tarefas. Sendo basicamente quatro tipos: uma máquina, máquinas em paralelo, máquinas em série e oficina de máquinas¹.

¹O emprego do termo máquina generaliza o recurso produtivo a ser utilizado: estação de trabalho, máquinas, mão-de-obra, etc.

- Máquina única (*single machine shop*): neste caso, consideram-se “ n ” tarefas a serem executadas em uma única máquina. Cada tarefa tem seus parâmetros definidos, como tempo de *setup*, tempo de produção, data de entrega, etc. Não é o caso mais utilizado pois em situações reais, geralmente tem-se mais máquinas, mas pode-se aplicar os métodos de uma única máquina em pontos específicos da produção, como por exemplo, em gargalos. Ou ainda em uma escala macro, considerar uma linha inteira ou mesmo uma fábrica inteira seja tratada como uma única máquina em um planejamento maior.
- Máquinas em Paralelo (*parallel machine shop*): trata-se de uma generalização do *single machine shop*, onde agora consideram-se “ n ” tarefas a serem executadas em “ m ” máquinas semelhantes. O objetivo neste caso, é determinar quais tarefas serão alocadas a cada máquina, e para cada máquina, uma sequência de execução.
- Máquinas em Série ou Fluxo (*flow shop*): nesta situação, tem-se “ n ” tarefas a serem executadas sequencialmente em “ m ” máquinas, cada tarefa com seu tempo de execução determinado em cada uma das máquinas. Ainda pode-se subcategorizar duas situações:
 - Roteiros idênticos (*pure flow shop*): todas as tarefas passam por todas as etapas de produção. Por exemplo, todas as tarefas devem passar pelas máquinas 1, 2 e 3.
 - Roteiros variáveis (*general flow shop*): algumas tarefas podem “pular” alguma etapa de produção. Por exemplo, algumas tarefas devem passar pelas máquinas 1, 2 e 3; outras devem passar apenas pelas máquinas 1 e 2.
- Oficina de máquinas (*job shop*): esta situação apresenta a configuração mais geral da programação da produção, havendo “ n ” tarefas a serem executadas em “ m ” máquinas com roteiros de etapas variáveis. Ou seja, cada tarefa tem sua sequência própria, sempre com seus parâmetros definidos, possibilitando inclusive, a reentrada de uma tarefa em uma máquina pela qual ela já tenha passado.

Podem ser considerados alguns aspectos além dos mencionados anteriormente na programação da produção, dentre os quais:

- Tempo de preparação dependente da sequência (*sequence dependent setup*): quando o tempo de preparação de uma máquina muda em função da tarefa anteriormente executada nela, tem mais um fator complicador, pois a combinatoriedade do problema aumenta.
- Preempção (*preemption*): é a admissão de uma interrupção de uma tarefa para o processamento de outra, podendo ser com recomeço, onde a execução da tarefa interrompida é perdida, ou com continuação, quando a tarefa interrompida continua do ponto onde foi parada.

Colin (2007) representa as configurações de cada ambiente na figura 2.1, onde no caso da oficina de máquinas (*job shop*), não perdendo a generalidade, utilizou-se apenas 3 máquinas; e é empregado o termo *ordem* para as tarefas.

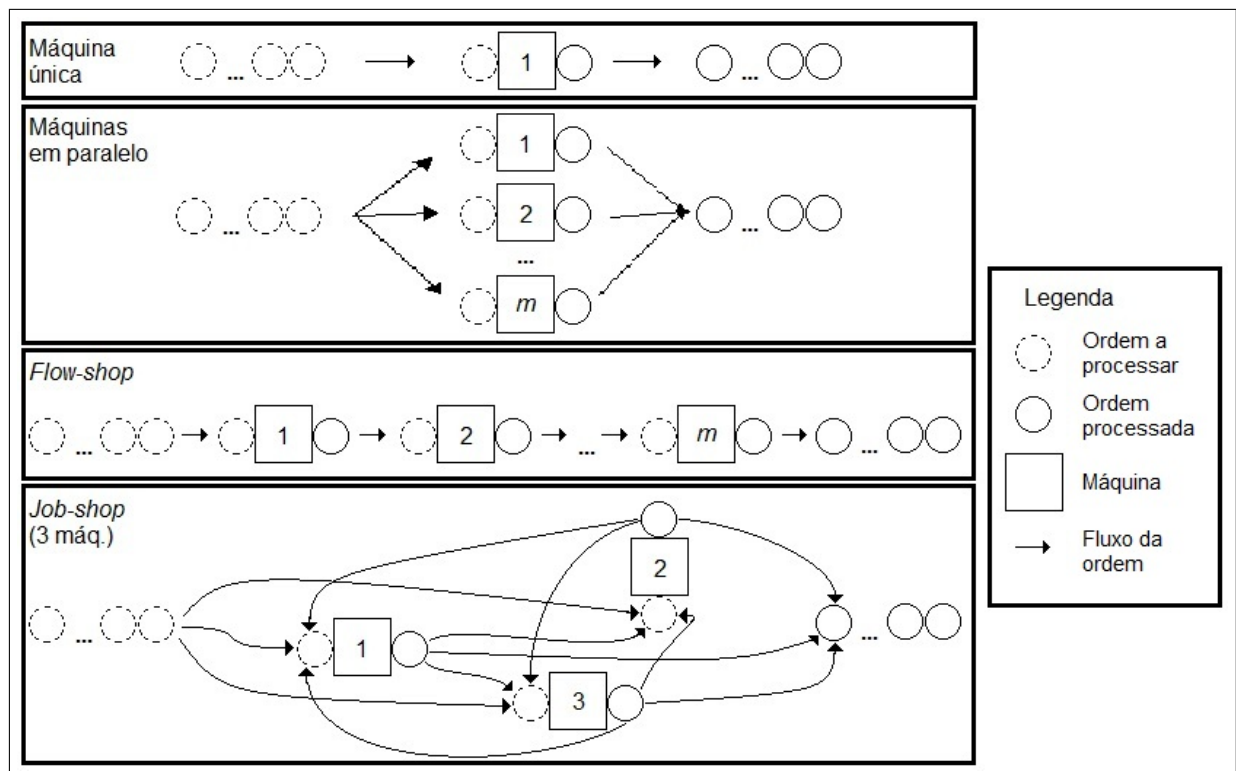


Figura 2.1: Configuração dos ambientes

Fonte: Colin (2007)

Pode-se observar que as configurações dos ambientes máquina única, máquinas em paralelo e o *flow shop* nada mais são que casos específicos do ambiente *job shop*.

2.1.3 Objetivos da Programação

Em geral, os objetivos da programação da produção são classificados em três classes: cumprimento de prazos, velocidade de fluxo e utilização.

- Cumprimento de prazos: trata diretamente do atendimento ao cliente.
- Velocidade de Fluxo: diz respeito ao tempo em que a tarefa permanece sendo executada na fábrica. Ponto este que pode gerar demora no chão de fábrica e implicar no não cumprimento dos prazos. Também, quanto maior o número de tarefas, maior o tempo de execução como um todo. Com a otimização deste item, busca-se a redução de filas de espera da execução das tarefas, aumentando a velocidade de fluxo.
- Utilização: busca a máxima utilização ou ociosidade mínima dos recursos, porém esta parte do objetivo normalmente é alcançado em um nível mais alto do planejamento do PCP.

Têm-se vários indicadores para avaliar a programação da produção, sendo alguns deles:

- Tempo total de trabalho (*makespan*): tempo total de execução de todas as tarefas consideradas na programação da produção.
- Tempo médio de fluxo (*mean flow time* ou *lead time*): média entre os tempos de fluxo das tarefas.
- Atraso médio ou máximo: média dos atrasos ou maior atraso entre as tarefas consideradas.
- Número de tarefas atrasadas: quantidade de tarefas concluídas depois da data de entrega acertada.

Defini-se então, o problema de programação e qual(is) o(s) critério(s) para avaliar a programação. Após esta etapa, escolhe-se o método que produza alternativas de solução para o problema da programação da produção.

Esta tese trata do tipo de decisão da Programação (*Scheduling*), no ambiente: Oficina de Máquinas (*Job Shop Scheduling*) considerando o tempo de preparação dependente da máquina e com o objetivo de minimizar o Tempo Total de Trabalho (*Makespan*).

2.1.4 Regras de Prioridade

Para certos problemas, em situações bem específicas, existem métodos que geram soluções exatas (ótimas), mas em geral, são apresentadas heurísticas. A seguir estão indicadas as regras que definem as heurísticas de mais comuns, com a sua sigla, nome e definição.

- mTP (SPT) - Menor Tempo de Processamento (*Shortest Processing Time*): sequenciam-se as tarefas pelo menor tempo de processamento.
- MTP (LPT) - Maior Tempo de Processamento (*Longest Processing Time*): sequenciam-se as tarefas pelo maior tempo de processamento.
- PEPS (FIFO) - Primeiro a Entrar, Primeiro a Sair (*First in, First Out*): sequenciam-se as tarefas pela ordem de chegada do seu pedido.
- MDE (EDD) - Menor Data de Entrega (*Earliest Due Date*): sequenciam-se as tarefas pela data de entrega prometida.
- TFM (MST) - Tempo de Folga Mínimo (*Minimum Slack Time*): sequenciam-se as tarefas pelo menor tempo de folga: diferença entre a data de entrega e o tempo de processamento.
- RC (CR) - Razão Crítica (*Critical Ratio*): sequenciam-se as tarefas com menor tempo remanescente: diferença entre a data de entrega e a data corrente.

Depois da regra de sequenciamento ter sido definida e executada para determinar o sequenciamento de todas as tarefas, uma função objetivo pode ser calculada para mensurar a qualidade do sequenciamento. Mas segundo Colin (2007), “não é comum a avaliação da função objetivo com o intuito de melhorar seu valor. Mesmo os chamados softwares de programação com capacidade finita não utilizam funções objetivo para melhorar os programas de produção”.

2.2 Revisão da literatura

Esta seção apresenta uma revisão da produção científica em torno do assunto. Os vários trabalhos pesquisados e estudados deram diferentes contribuições para o problema principal em termos de novos conceitos, modelos e métodos de solução.

Nas pesquisas feitas à literatura, muito se é dedicado a fornecer diferentes formulações para o problema de acordo com várias condições de trabalho que podem surgir na realidade. Um ponto importante a se considerar são as relações de precedência entre os processos necessários de um item, quando cada tarefa pode ser potencialmente executados em máquinas específicas (IERAPETRITOU e FLOUDAS, 1998; IERAPETRITOU, HENE e FLOUDAS, 1999).

Schwindt e Trautmann (2000) lidaram com o sequenciamento em lotes em processos industriais para o caso de máquinas com diferentes funções. Eles usaram métodos de sequenciamento através de restrição de recursos para lidar com o problema. Outras contribuições semelhantes são propostas na literatura, integrando ideias de outros problemas, como Wong, Chan e Chan (2009), que combinaram ideias de dimensionamento de lotes e o problema de sequenciamento com o problema de restrição de recursos. Além disso, eles assumiram que a sequência de encaminhamento das tarefas entre máquinas é conhecida *a priori*.

Outra dificuldade nas formulações é a consideração de diferentes tipos de capacidades e/ou escassez. Ozturk e Orneck (2010) desenvolveram um modelo de programação linear inteira-mista (MIP) multi-período para a programação da produção, com restrição de capacidade de produção junto com lotes vinculados. Seu modelo é uma extensão do dimensionamento de lotes e o problema de sequenciamento (citado acima) para ambientes de *job shop*. Eles utilizaram o CPLEX² para resolver o modelo proposto.

O problema de *scheduling* é resolvido com diferentes métodos de solução na literatura que podem ser categorizados principalmente em modelos exatos (para problemas de tamanho menor ou simples) e (meta-) heurísticas ou métodos híbridos (para problemas

² IBM ILOG CPLEX Optimization Studio - software que contém pacotes de otimização.

complicados ou de tamanho grande). De modo geral, pode-se considerar o *flow shop* como um caso especial do *job shop* (como apresentado na Figura 2.1, página 10, onde as relações de precedência entre os processos / máquinas são lineares). Neste sentido, a programação da produção em *job shop* é invariavelmente mais difícil do que *flow shop*. Assim, a maioria dos problemas da literatura na área de *job shop* são resolvidos através de métodos que não são exatos, em contraste com os problemas em outros ambientes, tais como *flow shop* e máquinas individuais (única/paralelo).

Como exemplo, Kim e Kim (1996) e Wong *et al.* (2009) resolveram os seus respectivos problemas de *scheduling* por *Simulated Annealing* e Algoritmos Genéticos. Nozick *et al.* (2001) propôs um modelo simples para o processamento de tarefas em um ambiente de *job shop* com a ajuda de conceitos de projetos de programação. Eles resolveram o modelo por uma heurística baseada em GBD (*Generalized Benders Decomposition*). Cheung e Zhou (2001) apresentaram um algoritmo híbrido entre um Algoritmo Genético e uma Heurística para o problema no ambiente de *job shop* com tempo de *setup* variável, sendo a heurística uma subrotina do Algoritmo Genético.

Rocha *et al.* (2004) utilizaram a meta-heurística GRASP para obter um limitante da execução do algoritmo de *Branch and Bound* para resolver o modelo proposto para o problema de máquinas paralelas com tempo de preparação dependente da sequência.

Alvarez-Valdes *et al.* (2005) consideraram uma situação particular de *job shop* aplicada a uma fábrica de produto de vidro, levando em conta as características das máquinas e das diferentes formas de processo dos produtos. Eles consideraram como objetivo custos e o prazo de entrega dos produtos aos clientes. Assim como neste trabalho, e em diversos outros pesquisados, os problemas de *job shop* muitas vezes são apresentados em situações de problemas específicos.

Nowicki e Smutnicki (2005) apresentaram um algoritmo baseado na Busca Tabu para apresentar soluções para o *job shop*, em uma situação genérica, com o *makespan* como objetivo, porém, não consideraram o tempo de preparação das máquinas. Liu *et al.* (2006) consideraram o mesmo problema, mas propuseram um algoritmo genético melhorado baseado no algoritmo híbrido *Taguchi-genetic*, que busca uma melhoria entre as etapas de *crossover* e da mutação. Já Huang e Liao (2008) utilizaram um híbrido entre

a meta-heurística de Otimização por Colônia de Formigas e Busca Tabu.

Ruiz *et al.* (2005) consideraram o tempo de preparação das máquinas dependentes da sequência com o objetivo de otimizar o tempo total de trabalho, porém, no ambiente de *flow shop*, através de buscas locais de permutações das tarefas sequenciadas. Neste mesmo trabalho, eles citam que buscaram os melhores parâmetros para as meta-heurísticas utilizadas por delimitamento de experimentos. Tasgetiren *et al.* (2007) trabalharam o mesmo problema, mas sem o tempo de preparação das máquinas, utilizando a meta-heurística de Otimização por Nuvem de Partículas.

Tanto Algoritmos Genéticos quanto o *Simulated Annealing* utilizam-se de parâmetros para sua execução, e Sadegheih (2006) apresentou um estudo sobre a influência dos mesmos sobre os problemas de *flow shop* e *job shop*, de forma comparativa entre as duas formas de solução. As conclusões foram sobre como a velocidade de resfriamento do sistema influencia no *Simulated Annealing* e o quanto a amplitude de intervalo dos parâmetros de um Algoritmo Genético podem deteriorar (termo usado por Sadegheih) a qualidade das soluções, e ainda sobre o quanto os Algoritmos Genéticos são mais rápidos que o *Simulated Annealing*.

Monkman *et al.* (2008) apresentaram mais um trabalho em uma situação específica do problema de *job shop*, neste caso, uma indústria de produtos eletrônicos. Eles consideraram o tempo de preparação dependente da sequência, mas na forma de custo, e seu objetivo foi minimizar o custo com os tempos de preparação. Para tal, utilizaram-se de uma heurística que envolveu três etapas - designação, sequenciamento e tempo de programação (no conceito de *scheduling*) - através do GRASP.

Qian *et al.* (2008) consideraram o *job shop* de forma multi-objetivo levando em conta o tempo total de trabalho e o máximo atraso. Para resolver seu modelo, eles utilizaram a meta-heurística dos Algoritmos Meméticos baseados em Evolução Diferencial, tanto com busca global, quanto em busca adaptativa local.

Luo *et al.* (2009) apresentaram um problema em dois estágios de sequenciamento em *flow shop* por lotes em um caso específico de uma empresa de metalurgia. O primeiro estágio tem quatro máquinas em paralelo e o segundo, apenas uma máquina. Eles con-

sideraram itens específicos para a produção desta empresa, dentre os quais o tempo de preparação dependente da sequência. Para solucionar o modelo desenvolvido por eles, foi utilizado um Algoritmo Genético.

Manikas e Cheng (2009) trabalharam o problema de *job shop* de forma multi-critério no objetivo e realizaram testes com várias funções objetivo considerando ponderações entre o tempo total de trabalho, atraso máximo, entrega dos produtos, e resolveram os modelos por Algoritmos Genéticos. Eles citam a questão do tempo de processamento dependente da sequência no início do artigo, mas na descrição do modelo e nos exemplos executados, não fica claro se foi de fato considerado na resolução dos problemas citados, mais parecendo que não foram considerados, ou que os mesmos foram incorporados aos tempos de processamento das tarefas, muito embora este tempo seja dependente da tarefa executada anteriormente na máquina em questão.

Um problema de *job shop* com o objetivo de minimizar o atraso de entrega foi trabalhado por Zhou *et al.* (2009). Eles utilizaram um híbrido de Heurística e Algoritmo Genético da forma: o Algoritmo Genético determina qual tarefa deve ser processada por primeiro em cada máquina e a Heurística a designação de qual o resto da sequência das tarefas nas máquinas. Eles não consideraram o tempo de preparação dependente da sequência em seu modelo.

Parthanadee e Buddhakulsomsiri (2010) realizaram simulações de modelos e analisaram os resultados para determinar qual a regra de prioridade mais apropriada para um estudo de caso específico na indústria de frutas (na Tailândia), levando em conta características do problema como a incerteza de haver frutas boas e na quantidade necessária disponíveis, diferentes produtos finais que compartilham o mesmo insumo tornando o sequenciamento das tarefas interdependentes entre si. O interessante neste artigo é que em nenhum momento eles citam se o problema apresentado por eles está no ambiente de *flow shop* ou *job shop*, apesar de citarem trabalhos na revisão da literatura que trabalham em um ou outro ambiente.

Mati *et al.* (2011) apresentaram um modelo de busca local e troca de vizinhança de arcos disjuntos no grafo que o problema de *scheduling* pode ser representado, no ambiente *job shop*, buscando otimizar qualquer critério indicador de avaliação da produção

como os descritos na seção anterior. O modelo proposto é resolvido por uma heurística dependente de apenas um parâmetro, e não foi considerado o tempo de preparação das máquinas. Sels *et al.* (2011) também trabalharam o problema de *job shop*, mas através de um Algoritmo Genético e um procedimento de busca dispersa, trabalhando não apenas com uma população, mas com duas populações, e assim, com uma maior diversidade e qualidade de informação, para então efetuar as trocas entre indivíduos mais bem classificados das distintas populações, no intuito de adicionar diversidade no processo de pesquisa de soluções. Eles também buscaram mostrar vantagens nesta forma de Algoritmo Genético sobre algumas meta-heurísticas.

Georgiadis e Michaloudis (2012) utilizaram um sistema dinâmico para representar o problema de *job shop*, examinando um caso específico de chão de fábrica, em que os efeitos da chegada de pedidos feitos pelos clientes e os vários tipos de eventos que podem ocorrer na situação real em função destes pedidos e, também, de falhas de máquinas são considerados. Apesar do sistema apresentado por eles buscar uma proximidade maior entre a teoria e a realidade, o tempo de preparação dependente da sequência não é utilizado, no sentido literal de preparação e ajustes na máquina antes da tarefa efetivamente ser realizada na máquinas. Este apenas é citado como muitas vezes necessário, e no sistema deles é considerado apenas o tempo independente da sequência. O termo *setup* muitas vezes é empregado na execução em si da tarefa na máquina.

Shen e Buscher (2012) consideram o tempo de preparação da máquina dependente da sequência buscando minimizar o tempo total de trabalho no ambiente de *job shop*. Mas na intenção de otimizar o processo, é primeiro realizado o loteamento das tarefas, quase que como um agrupamento de tarefas com certo grau de semelhança, levando a uma diminuição considerável do número de tarefas efetivas a ser sequenciadas, para então aplicar um Algoritmo do tipo Busca Tabu e chegar à solução.

Finalmente, mais recentemente, Karimi-Nasab e Seyedhoseini (2013) propuseram um modelo de Programação Linear Inteira para efetuar loteamento das tarefas e seu sequenciamento, no ambiente *job shop*, e consideraram o tempo de preparação dependente da sequência, mas na forma de otimizar o custo financeiro destes tempos de preparação e não com o objetivo de minimizar o tempo total de trabalho. Foi incorporado ao modelo a

ideia de máquinas flexíveis que permitem o gerenciador da produção mudar a velocidade de trabalho das máquinas. Como o modelo tem diversas restrições de desigualdade, e estas reduzem o espaço de soluções, elas são tratadas com o que é chamado no trabalho de planos de corte (*cutting planes*). Estas são utilizadas para resolver o modelo em *cut and branch* e *branch and cut*. Eles executaram o modelo no CPLEX. Os mesmos citam nos resultados que o modelo tenta ser bem próximo do problema real, mas que para ser aplicado em situações de muitas tarefas e máquinas (em situações reais de fato), deve-se previamente estabelecer os planos de corte de forma a passar em pontos específicos do espaço de soluções, e que possivelmente outras formas de solução como decomposições ou relaxamento de Lagrange possam trazer melhores resultados no futuro.

A tabela 2.1 resume alguns dos artigos pesquisados para a construção desta tese. Eles estão indicados sobre: a utilização ou não do tempo de preparação das máquinas (tempo de *setup*), sendo Dep. quando esta restrição é considerada e Indep. quando não; objetivo(s): o tempo total de trabalho e o custo financeiro são os objetivos mais apresentados, então os outros possíveis objetivos estão resumido em outros; e o método de solução utilizado para otimizar seus respectivos problemas. Sendo que a lista de siglas está no início do trabalho.

Tabela 2.1: Resumo da literatura sobre a programação da produção

Autor(es)	Setup		Objetivo(s)			Método de Solução
	Dep.	Indep.	Makespan	Custo	Outros	
Giffler e Thompson (1960)		•	•			HE
Kim e Kim (1996)		•		•		SA+AG
Ierapetritou e Floudas (1998)		•			•	GAMS
Ierapetritou <i>et al.</i> (1999)		•			•	GAMS
Schwindt e Trautmann (2000)	•		•			HE
Cheung e Zhou (2001)	•		•			AG+HE
Rocha <i>et al.</i> (2004)	•				•	B&B+GRASP
Alvarez-Valdes <i>et al.</i> (2005)		•		•		HE
Nowicki e Smutnicki (2005)		•			•	BT
Ruiz <i>et al.</i> (2005)	•		•			MHE
Liu <i>et al.</i> (2006)		•	•			AG

(continua na próxima página)

Tabela 2.1: Resumo da literatura sobre a programação da produção - continuação

Autor(es)	Setup		Objetivo(s)			Método de
	Dep.	Indep.	Makespan	Custo	Outros	Solução
Sadegheih (2006)		•		•		AG+SA
Tasgetiren <i>et al.</i> (2007)		•	•			ONP+VNS
Huang e Liao (2008)		•	•			OCF+BT
Monkman <i>et al.</i> (2008)	•			•		GRASP
Qian <i>et al.</i> (2008)		•	•	•	•	MHE
Luo <i>et al.</i> (2009)	•		•			AG
Manikas e Chang (2009)		•			•	AG
Wong <i>et al.</i> (2009)		•		•		AG
Zhou <i>et al.</i> (2009)		•			•	AG + HE
Ozturk e Orneck (2010)		•		•		PIM
Parthanadee e Buddhakulsomsiri (2010)	•		•	•	•	SM
Mati <i>et al.</i> (2011)		•			•	HE
Sels <i>et al.</i> (2011)		•	•			AG
Georgiadis e Michaloudis (2012)		•			•	HE
Shen e Buscher (2012)	•		•			BT
Karimi-Nasab e Seyedhoseini (2013)	•			•		B&C

A pesquisa em artigos e publicações foi além da apresentada. Os artigos citados foram os considerados mais relevantes em cada área que se propuseram a trazer inovação científica. Porém, em nenhum momento desta vasta pesquisa, foi encontrado uma heurística independente de parâmetros subjetivos (dependentes da escolha do programador) que resolvesse de forma rápida e simples o problema de otimizar o tempo total de trabalho da programação da produção no ambiente *job shop* levando em conta o tempo de preparação dependente da sequência.

Para o desenvolvimento da heurística que será apresentada no próximo capítulo, foi estudado extensivamente a heurística apresentada por Giffler e Thompson (1960), trabalho até hoje considerado referência no assunto de otimização do tempo total de

trabalho em ambientes *job shop*. Porém, o tempo de preparação das máquinas não é considerado. Então, sobre este trabalho de 1960, é baseado o algoritmo desta tese.

2.3 Algoritmo de Giffler e Thompson

Nesta seção será apresentado o algoritmo que Giffler e Thompson desenvolveram em 1960. Apesar de ser um algoritmo com mais de 50 anos de existência, o mesmo é muito eficiente para a otimização do tempo total de trabalho em ambientes *job shop* sem restrições.

Como notação para este algoritmo, é utilizado:

- Cada tarefa deve seguir uma sequência pré-determinada de máquinas. Esta sequência chama-se sequência tecnológica, onde a cada máquina que esta tarefa deve passar, há um tempo de processamento associado.
- Para cada tarefa, em cada etapa da sua sequência tecnológica, defini-se como operação O_{ij} da tarefa i na máquina j o conjunto de informações sobre esta etapa, descritas a seguir.
- Como cada tarefa tem uma sequência tecnológica independente, então é apresentada a sequência tecnológica T_{ik} e associada a esta, tem-se a matriz com os tempos de processamento p_{ij} da tarefa i na máquina j .
- Uma operação O_{ij} tem uma tarefa e máquina antecessora. A operação antecessora nomeada $PJ(O_{ij})$ é a antecessora de O_{ij} na sequência tecnológica T_{ik} . A definição da máquina antecessora $PM(O_{ij})$ é a última operação sequenciada na máquina, ou seja, a operação que terminou com o mais tarde entre as operações já sequenciadas na mesma máquina.
- O tempo de início $ES(O)$ de O é definido como o tempo de conclusão máximo de $PJ(O)$ e $PM(O)$, ou seja, $ES(O) = \max\{c(PJ(O)), c(PM(O))\}$, onde $c(O)$ é o tempo de conclusão de O . O tempo de conclusão mais cedo $EC(O)$ é definido como $ES(O)$ mais seu tempo de processamento $p(O)$.

O algoritmo pode ser descrito da forma:

Passo 1: Iniciar G como um conjunto das operações que são as primeiras de cada tarefa na sequência tecnológica.

$$G = \{O_{1,T11}, O_{2,T21}, \dots, O_{n,Tn1}\}$$

Para cada operação $O_{ij} \in G$, fazer $ES(O_{ij}) := 0$ e $EC(O_{ij}) := p_{ij}$. Iniciar $S := \{ \}$.

Passo 2: Achar a operação que termina antes no conjunto G .

$$O_{i^*r} = \arg \min_i \{EC(O_{ij}) | O_{ij} \in G\}$$

Montar um subconjunto $G1$ de G com todas as operações a passar na máquina r .

$$G1 = \{O_{ij} \in G | j = r\}$$

Passo 3: Montar o conjunto conflito C subconjunto de $G1$ em que o tempo de início da operação $O_{ir} \in G1$ seja menor que o tempo de término da operação O_{i^*r} .

$$C = \{O_{ir} \in G1 | ES(O_{ir}) \leq EC(O_{i^*r})\}$$

Passo 4: Selecionar uma das operações do conjunto conflito C aleatoriamente, denominando esta operação como O_{kr} .

$$O_{kr} = rnd\{O \in C\}$$

Passo 5: Sequenciar O_{kr} na lista S que contém as operações já sequenciadas, com seus tempos de início e conclusão da operação.

$$S = S \cup O_{kr}$$

Passo 6: Atualizar o tempo de todas as outras operações em $G1$ da forma:

$$\begin{aligned} ES(O_{ir}) &= \max\{ES(O_{ir}); EC(O_{kr})\}, \forall O_{ir} \in G1 - \{O_{kr}\} \\ EC(O_{ir}) &= ES(O_{ir}) + p(O_{ir}), \forall O_{ir} \in G1 - \{O_{kr}\} \end{aligned}$$

Passo 7: Remover a operação sequenciada O_{kr} de G (e portanto de $G1$) e adicionar a próxima operação O_{kj} da sua sequência tecnológica (se ainda houver operações a sequenciar)

$$G = (G - \{O_{kr}\}) \cup \{O_{kj}\}$$

atualizando suas informações de tempo:

$$ES(O_{kj}) = \max\{EC(O_{kr}); EC(PM(O_{kj}))\}$$

$$EC(O_{kj}) = ES(O_{kj}) + p(O_{kj})$$

Passo 8: Repetir do **Passo 2** ao **Passo 7** até que todas as operações de todas as tarefas estejam sequenciadas.

Passo 9: Apresentar a lista S como um sequenciamento.

Com este algoritmo é obtida uma solução factível para o problema. Porém, esta solução pode estar longe de um ótimo, mesmo que local. Então, colocando o algoritmo apresentado como um processo iterativo, o mesmo pode ser repetido K vezes, sempre comparando-se a solução obtida na iteração atual com o melhor resultado já obtido. Ao término das K repetições, a melhor solução possível é encontrada, em função da aleatoriedade na escolha da operação a ser sequenciada no **Passo 4**.

Metodologia

Neste capítulo são apresentados: o algoritmo heurístico para otimizar o tempo total de trabalho na programação da produção em oficina de máquinas com tempo de preparação dependente da sequência, construído a partir das ideias do algoritmo apresentado por Giffler e Thompson (1960); a implementação do algoritmo proposto, com a linguagem de programação utilizada e estrutura dos arquivos de dados de entrada e saída; e um modelo exato de Programação Inteira Mista para a resolução de um problema tratado nesta tese para validação dos resultados obtidos pela heurística.

3.1 O Algoritmo

Para a descrição do algoritmo proposto, o mesmo utiliza-se das seguintes notações:

- i - tarefa
- j - máquina
- T_{ik} - Matriz tecnológica representando a tarefa i na etapa k da sua sequência, ainda $T_{ik} = j$
- O_{ij} - Operação da tarefa i a ser executada na máquina j
- $TI(O_{ij})$ - Tempo de início da execução da tarefa i na máquina j
- $TC(O_{ij})$ - Tempo de conclusão (término) da execução da tarefa i na máquina j
- $TP(O_{ij})$ - Tempo de preparação (*setup*) para a tarefa i poder ser executada na máquina j
- $TE(O_{ij})$ - Tempo de execução da tarefa i na máquina j
- $MC(O_{ij})$ - Tempo de conclusão mais cedo da tarefa i na máquina j

- Γ_j - Tempo de conclusão da última tarefa sequenciada na máquina j
- Φ_j - Última tarefa sequenciada na máquina j
- $\Sigma_{j,\Phi_j,i}$ - Matriz de tempo de preparação da máquina j , com tarefa Φ_j , com a próxima tarefa a ser executada i
- Ω - Lista com as tarefas sequenciadas pelo algoritmo

O algoritmo proposto pode ser apresentado segundo os seguintes passos:

Passo 1: Iniciar G como um conjunto das operações que são as primeiras de cada tarefa na sequência tecnológica.

$$G = \{O_{1,T11}, O_{2,T21}, \dots, O_{n,Tn1}\}$$

Para cada operação $O_{ij} \in G$, fazer $TI(O_{ij}) := 0$. Iniciar $S := \{ \}$.

Passo 2: Achar a operação que termina mais cedo no conjunto G , considerando o tempo de preparação necessário para a tarefa entrar na máquina, em função da última operação sequenciada na mesma.

$$O_{i^*r} = \arg \min_i \{MC(O_{ij}) | O_{ij} \in G\}$$

onde

$$MC(O_{ij}) = \begin{cases} \Gamma_j + TE(O_{ij}) + \Sigma_{j,\Phi_j,i} & , \text{ se } \Gamma_j \geq TI(O_{ij}) \\ \Gamma_j + TE(O_{ij}) + (\Gamma_j + \Sigma_{j,\Phi_j,i} - TI(O_{ij})) & , \text{ se } \Gamma_j + \Sigma_{j,\Phi_j,i} \geq TI(O_{ij}) \\ \Gamma_j + TE(O_{ij}) & , \text{ caso contrário.} \end{cases}$$

Passo 3: Montar o conjunto conflito C , subconjunto de G , em que o tempo de início da operação $O_{ir} \in G$ mais o tempo de preparação necessário para executar a tarefa i na máquina r seja menor que o tempo de término da operação O_{i^*r} .

$$C = \{O_{ir} \in G | MC(O_{ir}) - TE(O_{ir}) \leq MC(O_{i^*r})\}$$

Passo 4: Selecionar uma das operações do conjunto conflito C aleatoriamente, denominando esta operação como O_{xr} .

$$O_{xr} = rnd\{O_{ir} \in C\}$$

Passo 5: Sequenciar O_{xr} na lista Ω que contém as operações já sequenciadas O_{ij} , com as informações de tempo de preparação $TP(O_{ij})$, tempo de conclusão $TC(O_{ij})$ e tempo de início $TI(O_{ij})$ da operação.

$$\Omega = \Omega \cup \{O_{xr}\}$$

onde

$$TP(O_{xr}) = \Sigma_{r, \Phi_r, k}$$

$$TC(O_{xr}) = MC(O_{kr})$$

$$TS(O_{xr}) = MC(O_{kr}) - TE(O_{kr})$$

Passo 6: Atualizar o tempo de todas as outras operações O_{ir} em G da forma:

$$TI(O_{ir}) = \max\{TI(O_{ir}); MC(O_{xr})\}, \forall O_{ir} \in (C - \{O_{xr}\})$$

Passo 7: Remover a operação sequenciada O_{xr} de G e adicionar a próxima operação O_{xj} da sua sequência tecnológica (se ainda houver operações a sequenciar).

$$G = (G - \{O_{xr}\}) \cup \{O_{xj}\}$$

atualizando sua informação de tempo inicial:

$$TI(O_{kj}) = \max\{TC(O_{kr}); TC(\Phi_j)\}$$

Passo 8: Repetir do **Passo 2** ao **Passo 7** até que todas as operações de todas as tarefas estejam sequenciadas.

Passo 9: Apresentar a lista Ω como um sequenciamento.

Pode-se perceber que a estrutura geral do algoritmo assemelha-se muito ao algoritmo de Giffler e Thompson (1960), porém agora é levado em conta a informação de decisão sobre o tempo de preparação das máquinas em função da tarefa anteriormente sequenciada.

O algoritmo inicia com a mesma construção de um conjunto, denotado por G , com as operações inicialmente sequenciáveis de cada tarefa. Então, diferentemente de Giffler e Thompson que apenas preocupam-se com o tempo de execução da operação na máquina em questão, o algoritmo proposto busca a informação sobre o tempo de preparação necessário neste momento para cada tarefa em cada máquina. O valor de $MC(O_{ij})$ calculado foi estruturado pela lógica apresentada a seguir.

Suponha que a tarefa i esteja sequenciada na máquina $j1$ e então na sua sequência tecnológica ela deva ser sequenciada na máquina $j2$. Têm-se então cinco cenários possíveis, denotados por $j2.k, k = 1, \dots, 5$, como apresentado na figura 3.1 (página 26), em função do tempo de preparação da tarefa h já sequenciada na máquina $j2$.

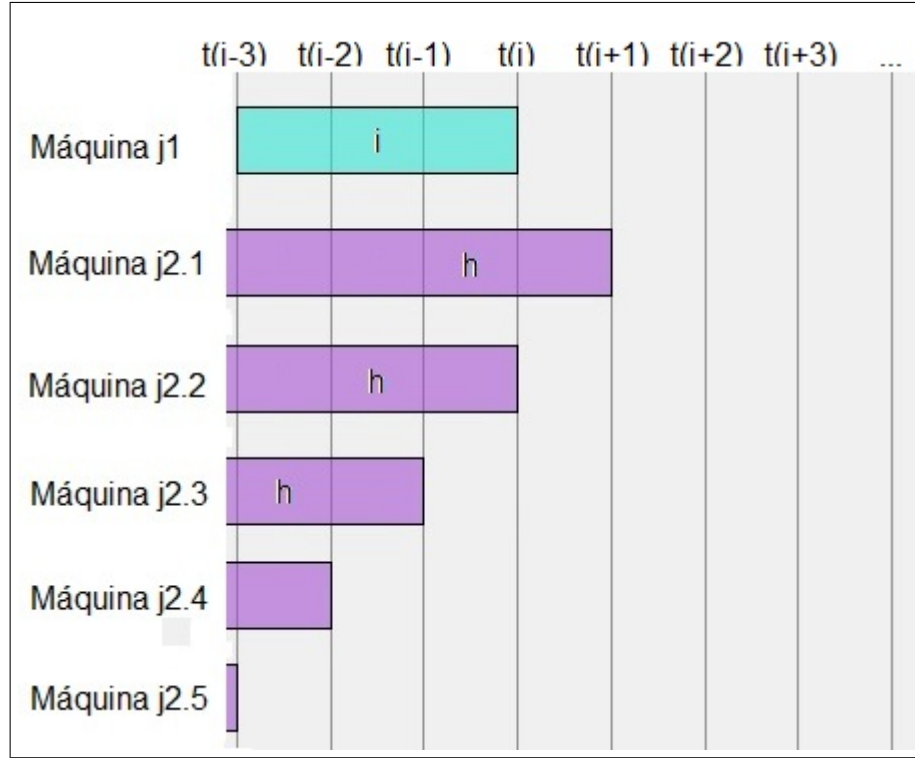


Figura 3.1: Cenários para a Etapa 3

Fonte: O autor

Para esta explicação, será considerado $\Sigma_{j,\Phi_j,i} = 2$. A primeira expressão para $EC(O_{ij})$ está condicionada a $\Gamma_j \geq TI(O_{ij})$, que são representadas nas condições $j2.1$ e $j2.2$ apresentadas na figura 3.1 (página 26), ou seja, quando o tempo da última tarefa sequenciada na máquina $j2$ é maior ou igual ao tempo de início da tarefa i - neste caso no tempo $TI(O_{ij}) = t(i)$. Se a tarefa encontra-se em qualquer uma destas duas condições, o

tempo de preparação $\Sigma_{j,\Phi_j,i}$ deve ser integralmente considerado para então iniciar a tarefa O_{ij} e seu tempo de conclusão $MC(O_{ij})$ ser calculado, conforme a figura 3.2 (página 27).

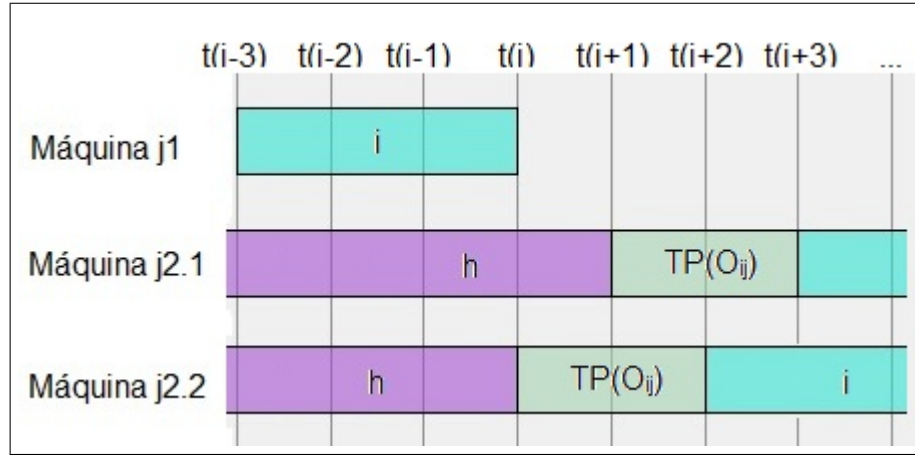


Figura 3.2: Cenários 1 e 2 para a Etapa 3

Fonte: O autor

A segunda expressão para $MC(O_{ij})$ está condicionada a $\Gamma_j + \Sigma_{j,\Phi_j,i} \geq TI(O_{ij})$, que são representadas nas condições $j2.3$ e $j2.4$ apresentadas na figura 3.1 (página 26), levando em conta $\Sigma_{j,\Phi_j,i} = 2$, ou seja, quando o tempo de conclusão da última tarefa sequenciada na máquina $j2$ é menor que o tempo de início da tarefa i , $TI(O_{ij})$. Então a consideração do tempo de preparação se faz necessária, conforme a figura 3.3 (página 27).

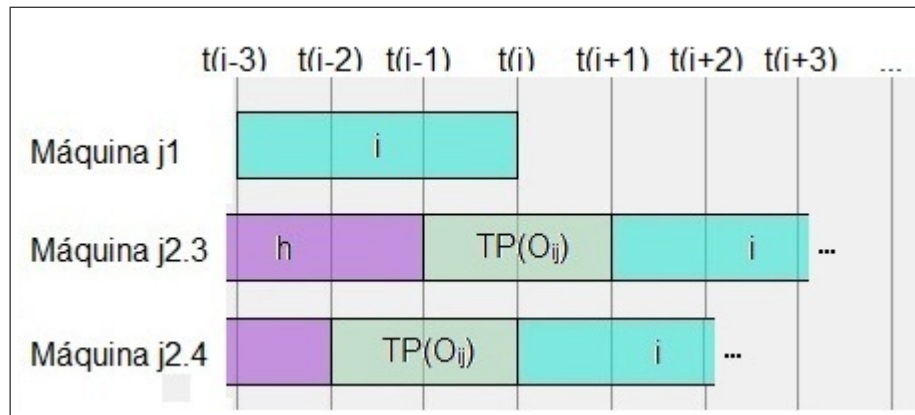


Figura 3.3: Cenários 3 e 4 para a Etapa 3

Fonte: O autor

Pode-se notar na figura 3.3 que há a possibilidade de, dado o tempo de preparação, o tempo inicial da tarefa ter que ser deslocado em uma quantidade de tempo menor que o próprio tempo de preparação em si. Por isso que a expressão para o cálculo de $MC(O_{ij})$

tem o termo $(\Gamma_j + \Sigma_{j,\Phi_j,i} - TI(O_{ij}))$, que representa a diferença entre, o tempo de conclusão da última tarefa na máquina $j2$ (neste caso) somado ao tempo de preparação $\Sigma_{j,\Phi_j,i}$, e o tempo de início $TI(O_{ij})$ da tarefa i . Esta quantidade de tempo está destacada em vermelho na figura 3.4 (página 28).

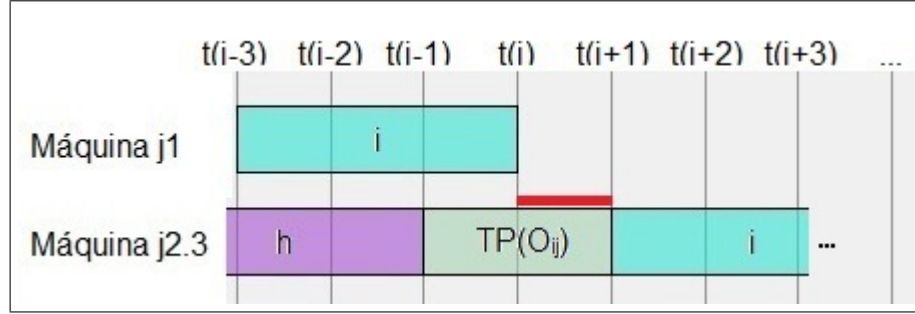


Figura 3.4: Cenário 3 para a Etapa 3

Fonte: O autor

A terceira expressão para $MC(O_{ij})$ é o caso que sobra, e está representado na condição $j2.5$, quando o tempo de conclusão da última tarefa sequenciada na máquina $j2$ somado ao tempo de preparação da tarefa i ainda é menor que o próprio tempo de início da tarefa i , figura 3.5 (página 28).

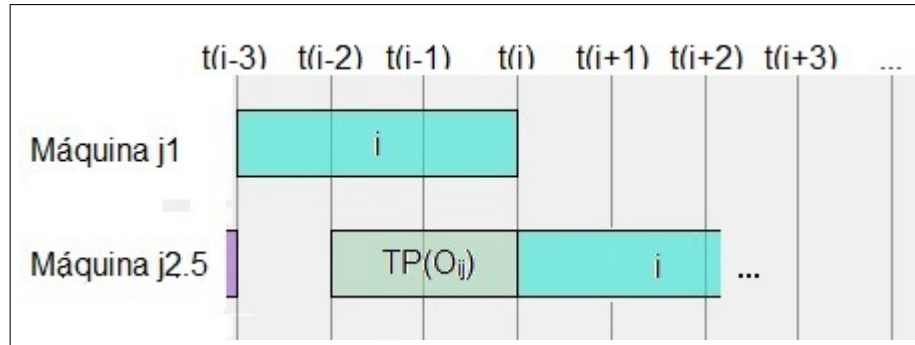


Figura 3.5: Cenário 5 para a Etapa 3

Fonte: O autor

Após estes dois primeiros passos, tem-se definido em qual máquina será estabelecido o próximo sequenciamento. É construído então o conjunto conflito C com as possíveis tarefas a sequenciar nesta máquina mas que, em função da tarefa que termina mais cedo nesta máquina, devem ter seu tempo início $TI(O_{ij})$, mais o tempo de preparação necessário, menor que o tempo de conclusão da tarefa i^* selecionada no passo anterior.

Formado o conjunto conflito, seleciona-se uma operação aleatoriamente para ser sequenciada em Ω , para garantir a diversidade de soluções possíveis ao fim de uma iteração.

Em função da tarefa sequenciada no passo anterior, atualizam-se as informações de tempo de início $TI(O_{ij})$ de todas as outras operações que estão no conjunto conflito C , mas diretamente em G . Remove-se então de G a tarefa sequenciada em Ω no passo anterior e, se ainda houver na sequência tecnológica da tarefa alguma operação a ser sequenciada, adiciona-se esta operação a G , atualizando o tempo de início da mesma em função da última operação sequenciada ou do tempo de conclusão da última operação sequenciada na máquina em que a nova operação deve ser sequenciada.

Repete-se os passos acima descritos até que todas as operações, de todas as tarefas tenham sido sequenciadas e apresenta-se Ω como a lista de sequenciamento das tarefas, todas levando em conta o tempo de preparação da máquina em função da última tarefa sequenciada na mesma.

Percebe-se a diferença de complexidade entre comparar apenas um tempo de execução da tarefa em cada máquina e, considerar além deste tempo, o tempo de preparação das máquinas, em função da última tarefa sequenciada e ter que avaliar qual o tempo de preparação efetivo a ser utilizado em cada máquina, o que diferencia o algoritmo proposto do apresentado por Giffler e Thompson, tornando-o de abrangência muito maior e maior aplicabilidade.

3.2 A Implementação

Nesta seção é apresentado o programa desenvolvido para a execução do algoritmo apresentado na seção anterior. A linguagem de programação utilizada para desenvolver a implementação do algoritmo desta proposta de tese foi o *Visual Basic 2012* inserido no *Microsoft Visual Studio Ultimate 2012 - Version 11.0.50727.1 RTMREL*. O nome dado ao programa desenvolvido é *FBN_JobShopScheduling_Tese* e sua interface com o usuário é apresentada na Figura 3.6 (página 30).

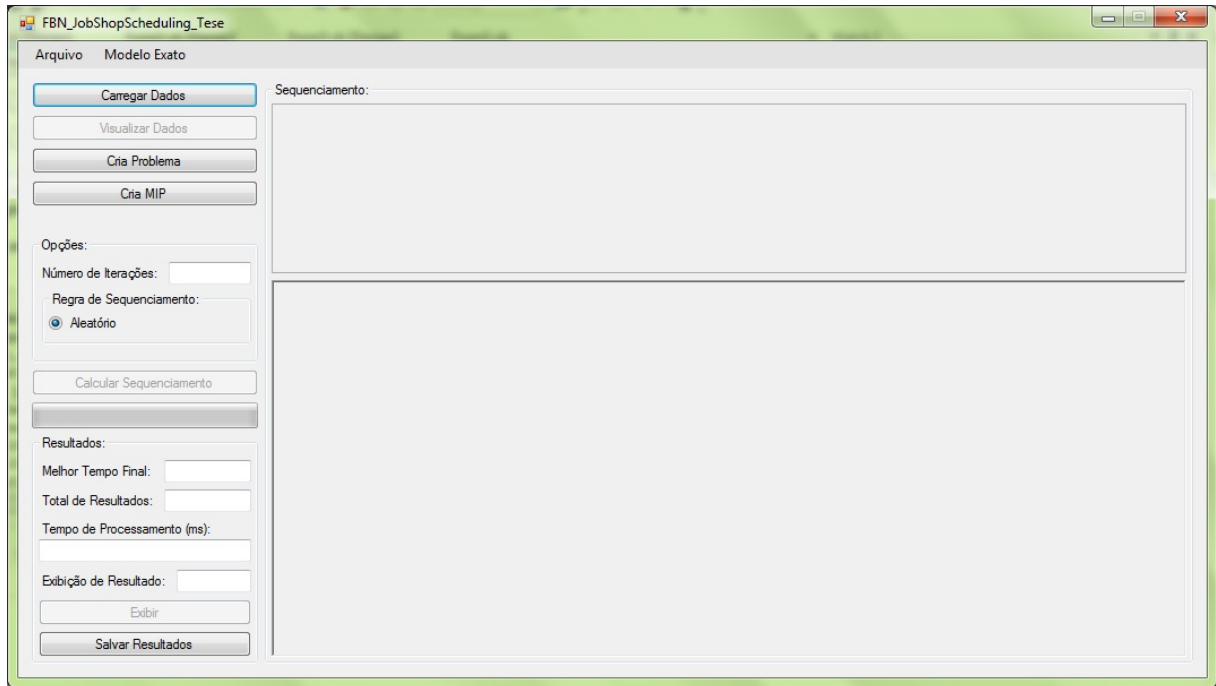


Figura 3.6: Interface com o usuário *FBN_JobShopScheduling_Tese*

Fonte: O autor

Como dados de entrada são necessárias as seguintes informações que descrevem o problema:

Número de Máquinas - total de máquinas utilizadas no problema;

Número de Tarefas - total de tarefas a serem executadas nas máquinas n ;

Matriz Tecnológica - esta informação deve ser passada na forma de um arranjo de n listas. O número de informações em cada lista pode variar, pois uma tarefa pode não ter que necessariamente passar por todas as máquinas, ou ainda pode ter que passar por uma mesma máquina mais de uma vez. Cada elemento da lista informa a operação da tarefa i : máquina j com tempo de execução na máquina $TE(O_{ij})$, na forma: $j; TE(O_{ij})$;

Matriz de Preparação - para cada máquina considerada, deve ser informado uma matriz $n \times n$, onde cada elemento i, j representa o tempo de preparação para a tarefa j poder ser executada na máquina, dado que a tarefa i foi a última a ser sequenciada na mesma.

A seguir um exemplo de arquivo com os dados de entrada do programa.

```
NumeroDeMaquinas: 3
```

```
NumeroDeTarefas: 3
```

```
Tarefas:
```

```
1;3 2;2 3;2
```

```
2;1 1;4 3;3
```

```
3;2 2;4 1;1
```

```
MatrizDeSetup
```

```
Maquina:1
```

```
0 1 2
```

```
2 0 3
```

```
1 4 0
```

```
Maquina:2
```

```
0 2 1
```

```
1 0 4
```

```
3 1 0
```

```
Maquina:3
```

```
0 3 2
```

```
3 0 1
```

```
3 2 0
```

```
EOF
```

Para a otimização da programação da produção, é necessário carregar o arquivo contendo as informações descritas anteriormente. Após a leitura dos dados, o programa, cria uma lista de listas com as sequências tecnológicas, onde cada elemento de cada lista é uma *struct* da forma:

```
Structure Opera
```

```
Dim tarefa As Integer
```

```
Dim maquina As Integer
```

```
Dim tempoStart As Single
```

```
Dim tempoSetup As Single
Dim tempoSetupAux As Single
Dim tempoComplete As Single
Dim tempoExec As Single
End Structure
```

onde Opera é o nome da variável *struct* criada para armazenar todas as informações relativas a cada operação de cada tarefa. Após a construção das estruturas com as informações da sequência tecnológica, são carregadas as matrizes com os tempos de preparação (*setups*).

Depois de todos os dados carregados, o usuário deve informar quantas vezes (iterações) deseja que o programa execute o algoritmo. Caso não seja informado um valor maior que zero, uma mensagem de erro é mostrada. Inserida a quantidade de iterações, ao clicar o botão “Calcular Sequenciamento”, o mesmo é processado, conforme os passos descritos na seção anterior, até chegar ao final de todas as iterações informadas pelo usuário. Ao final da execução das iterações, o resultado é apresentado conforme a figura 3.7 (página 33).

As informações sobre o sequenciamento da produção são apresentados de forma tabular, onde cada coluna da tabela é uma operação sequenciada, e a ordem de apresentação destas colunas é exatamente a ordem em que as mesmas foram sequenciadas. Em uma dada coluna, em cada linha tem-se a informação sobre (de cima para baixo): Máquina, Tarefa, Tempo de início de execução da tarefa (Tempo Start), Tempo de preparação (Tempo *Setup*) e Tempo final da execução da tarefa (Tempo Complete). Também é apresentado o sequenciamento das tarefas em um gráfico de Gantt, onde as barras em vermelho são os tempos de preparação entre as tarefas.

Como saída de dados, ao clicar em “Salvar Resultados”, é gerado um arquivo texto (*.txt*) com as informações:

Melhor resultado - tempo ótimo obtido pelo algoritmo;

Número de Iterações - total de iterações processadas;

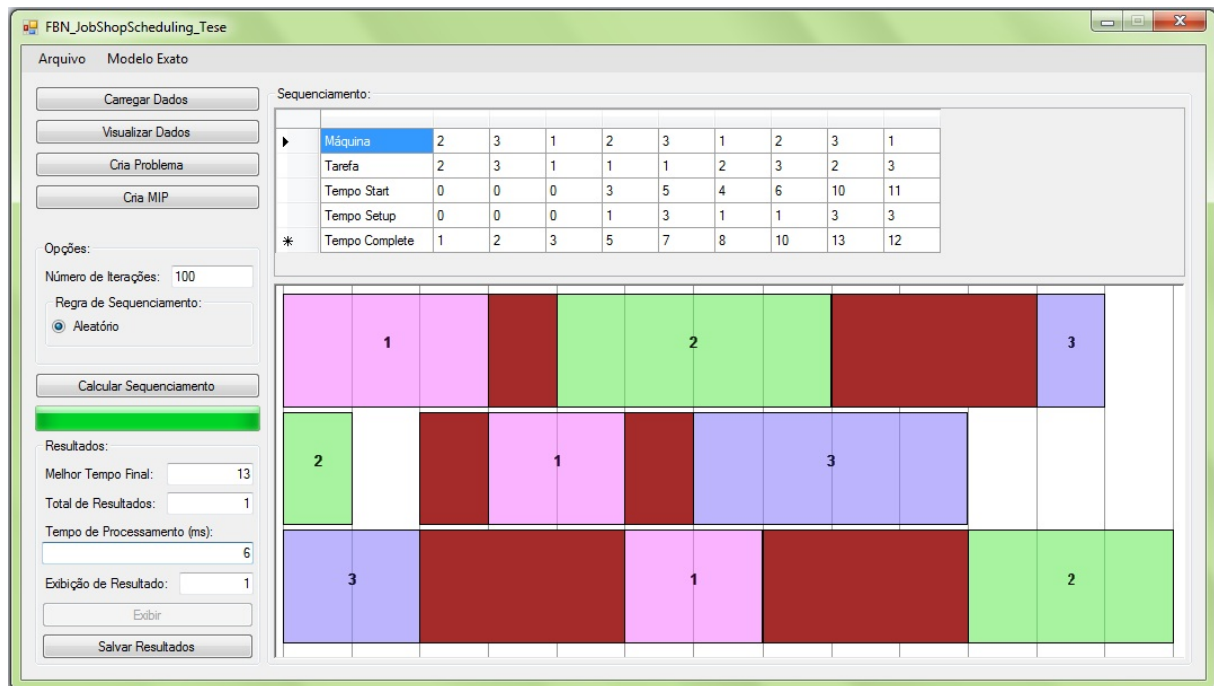


Figura 3.7: Interface de apresentação de resultados

Fonte: O autor

Número de Resultados - quantidade k de sequenciamentos diferentes com o mesmo valor de tempo total de trabalho

Resultado - resultado l para $l = 1, 2, \dots, k$, seguido de uma lista de valores (total de operações) com as informações na seguinte ordem: Máquina; Tarefa; Tempo Start; Tempo *Setup*; Tempo Complete.

Logo abaixo, a forma do arquivo com a saída de dados obtida na resolução do exemplo apresentado nesta seção e ilustrado na figura 3.7 (página 33).

MelhorResultado:13

NumeroDeIterações:100

Tempo Total de Execução (ms): 6

NumeroDeResultados:1

Resultado:1

2;2;0;0;1

3;3;0;0;2

1;1;0;0;3
2;1;3;1;5
3;1;5;3;7
1;2;4;1;8
2;3;6;1;10
3;2;10;3;13
1;3;11;3;12
EOF

O gráfico de Gantt também é salvo no formato de imagem (*.bmp*). Também são gerados, um arquivo do tipo *.lp* com o modelo de programação inteira mista apresentado na seção a seguir, e um arquivo de solução (*.sol*) com a solução obtida pela heurística no formato padrão de leitura do Cplex. Os arquivos para o exemplo utilizado nesta seção estão no Anexo A.

3.3 O Modelo de Programação Matemática

Para validação dos resultados obtidos pela heurística aqui apresentada foi considerado o modelo de Programação Inteira Mista (PIM) apresentado nesta seção.

Tal modelo tem como limitação, a necessidade de que cada tarefa passe por todas as máquinas uma e apenas uma vez. Este modelo foi adaptado do apresentado em Cheung e Zhou (2001).

O modelo utiliza-se das seguintes notações:

- n - número de tarefas
- m - número de máquinas
- i - tarefa i , $i = 1, 2, \dots, n$
- j - máquina j , $j = 1, 2, \dots, m$
- M - constante positiva suficientemente grande

Parâmetros:

- p_{ji} - tempo de processamento da tarefa i na máquina j ; $i = 1, 2, \dots, n$; $j = 1, 2, \dots, m$
- σ^i - sequência tecnológica da tarefa i , isto é, a ordem das máquinas que a tarefa i deve passar, $\sigma^i = (\sigma_1^i, \sigma_2^i, \dots, \sigma_m^i)$, $\sigma_j^i \in \{1, 2, \dots, m\}$ é o índice da correspondente máquina.
- s_{ik}^j - tempo de preparação requerido para processar a tarefa k na máquina j imediatamente após a tarefa i ter sido processada na máquina j , $j = 1, 2, \dots, m$; $i, k = 1, 2, \dots, n$

Variáveis:

- C_{\max} - *makespan* da programação da produção
- t_{ji} - tempo de início do processamento da tarefa i na máquina j
- $x_{ik}^j = \begin{cases} 1 & \text{se a tarefa } k \text{ é sequenciada imediatamente após } i \text{ na máquina } j, \\ 0 & \text{caso contrário} \end{cases}$

Modelo:

$$\min C_{\max}$$

Sujeito a:

$$t_{\sigma_h^i i} + p_{\sigma_h^i i} \leq t_{\sigma_{h+1}^i i}, \quad h = 1, 2, \dots, m-1; i = 1, 2, \dots, n \quad (3.1)$$

$$t_{\sigma_m^i i} + p_{\sigma_m^i i} \leq C_{\max} \quad i = 1, 2, \dots, n \quad (3.2)$$

$$t_{ji} + p_{ji} + s_{ik}^j x_{ik}^j \leq t_{jk} + (1 - x_{ik}^j)M \quad \begin{matrix} j = 1, 2, \dots, m; i = 1, \dots, n, \\ k = 1, 2, \dots, n, k \neq i \end{matrix} \quad (3.3)$$

$$\sum_{\substack{k=0 \\ k \neq i}}^n x_{ik}^j = 1, \quad j = 1, 2, \dots, m; i = 0, 1, \dots, n \quad (3.4)$$

$$\sum_{\substack{i=0 \\ i \neq k}}^n x_{ik}^j = 1, \quad j = 1, 2, \dots, m; k = 0, 1, \dots, n \quad (3.5)$$

$$x_{ik}^j + x_{ki}^j \leq 1, \quad j = 1, 2, \dots, m; i, k = 0, 1, 2, \dots, n, i \neq k \quad (3.6)$$

Sobre as restrições:

- (3.1) garante que a sequência tecnológica de cada tarefa seja respeitada,
- (3.2) é uma restrição natural em que todas as operações devem ser completas antes do *makespan*,
- (3.3) garante que a máquina só pode começar a processar a próxima tarefa apenas quando a operação prévia e o tempo de *setup* estejam completos,
- (3.4) na máquina j quando uma tarefa complete seu processamento, uma e apenas uma tarefa será selecionada para ser sequenciada após esta, a menos que a máquina j complete todas as operações,
- (3.5) implica em que a operação de uma tarefa deve ter uma e apenas uma tarefa predecessora, exceto quando a mesma é a primeira operação, e
- (3.6) garante que uma tarefa não seja sequenciada após outra que seja sua antecessora.

Este modelo tem $(m-1)n + n + mn(n-1) + 2m(n+1) + \frac{m}{2}n(n+1)$ restrições, mn variáveis positivas (t_{ji}) e $mn(n+1)$ variáveis binárias (x_{ik}^j). Assim, para um problema 10x10, por exemplo, o modelo terá 1770 restrições, 100 variáveis positivas e 1100 variáveis binárias, o que torna este modelo de difícil execução para problemas grandes.

Resultados

Neste capítulo são apresentados os testes realizados com o programa *FBN-JobShopScheduling-Tese*. O computador utilizado é um notebook HP Pavilion dm4-1075br, com processador Intel(R) Core(TM) i5 CPU 2.27GHz, 4,00GB de memória RAM. Para validar os resultados obtidos foram realizadas comparações entre os resultados encontrados pelo modelo matemático (PIM) apresentado na seção 3.3 e os obtidos pela heurística apresentada neste tese.

Todos os exemplos tem a mesma quantidade de máquinas e tarefas, além do fato de que todas as tarefas devem passar por todas as máquinas uma única vez, sendo esta uma limitação do modelo apresentado.

A heurística desenvolvida nesta tese contempla mais possibilidade de problemas, como a possibilidade de reentrada de uma tarefa em uma máquina pela qual ela já tenha sido processada, e a não necessidade de todas as tarefas passarem obrigatoriamente por todas as máquinas.

4.1 Problemas 3x3 a 7x7

Para realizar a validação dos resultados, foram gerados e testados 30 exemplos para cada tamanho do problema. Em cada exemplo, na heurística são salvos o melhor resultado obtido (fixado o número de iterações) e o tempo de processamento para obter tal solução, e no modelo matemático, resolvido no Cplex, foi salvo o ótimo obtido pelo mesmo, e o tempo de processamento para obter tal ótimo.

Em cada teste realizado, depois de salvas as informações, foram comparados os resultados do melhor tempo obtido pela heurística em relação ao ótimo obtido pelo método exato, pelas relações:

- $GAP_{solução} = \frac{\text{Melhor resultado heurística}}{\text{Melhor Solução PIM}} - 1$

Este resultado mostra o percentual da solução obtida pela heurística em relação à melhor solução obtida pelo modelo matemático. Por exemplo, um $GAP_{solução} = 2,94\%$ indica que o melhor resultado obtido pela heurística para um certo exemplo é 2,94% maior que o valor obtido pelo modelo matemático.

- $GAP_{tempo} = \frac{\text{Tempo heurística}}{\text{Tempo PIM}}$

Sendo que este resultado é o percentual de tempo que a heurística levou para obter a melhor solução em relação ao tempo que o modelo matemático levou para obter a melhor solução. Por exemplo, um $GAP_{tempo} = 4,55\%$ indica que a heurística levou 4,55% do tempo que o modelo matemático para obter a melhor solução.

Para cada tamanho de problema, de 3x3 a 7x7, foram contruídas as tabelas que encontram-se no Anexo B (página 57). As médias dos tempos de processamento (em ms) dos exemplos para cada ordem estão na tabela 4.1 a seguir.

Tabela 4.1: Tempos médios de processamento (ms) 3x3 a 7x7

Fonte: O autor

Tamanho	Iterações	Heurística	PIM
3x3	100	5,93	63,00
4x4	500	59,90	133,67
5x5	1000	215,53	864,57
6x6	5000	1718,63	32558,67
7x7	5000	2688,57	1628035,67

Então construiu-se o intervalo de confiança (de 95%) sobre os tempos médios de processamento, obtendo-se os resultados da Tabela 4.2 (página 39). Estes resultados comprovam haver diferença significativa entre os tempos de processamento da heurística e do modelo matemático, mesmo o número de iterações crescendo à medida que a ordem dos problemas também crescem.

Tabela 4.2: Intervalos de Confiança - tempos médios de processamento (ms) 3x3 a 7x7

Fonte: O autor

Tamanho	Heurística	PIM
3x3	(5, 52; 6, 34)	(42,80 ; 83,20)
4x4	(56, 84; 62, 96)	(104,14 ; 163,20)
5x5	(203, 42; 227, 65)	(637,40 ; 1091,74)
6x6	(1660, 81; 1776, 46)	(15851,19 ; 49266,14)
7x7	(2608, 91; 2768, 22)	(752033,79 ; 2504037,54)

Em cada um dos tamanhos processados nesta seção, após todos os testes realizados e salvos todas as informações necessárias, foram tomadas as médias dos $GAPs$ para as soluções e os tempos de processamentos, em cada ordem de problema. Os mesmo estão resumidos na Tabela 4.3 (página 39) e estes resultados podem ser visualizados graficamente na Figura 4.1 (página 40).

Tabela 4.3: Comparativos resultados 3x3 a 7x7

Fonte: O autor

Tamanho	$GAP_{solução}$	GAP_{tempo}
3x3	0,40%	15,26%
4x4	1,57%	52,82%
5x5	1,17%	39,03%
6x6	3,98%	12,15%
7x7	5,29%	1,51%

4.2 Problemas 8x8 e 9x9

Ao iniciar os testes para os problemas de ordem 8x8, observou-se que o tempo de resolução do modelo matemático para obtenção do ótimo tornou-se inviável, passando de 24 horas o tempo de processamento e ainda assim o ótimo não era atingido. Então, para continuar com a validação dos resultados, foram comparados os melhores valores obtidos pela heurística e a solução que o modelo matemático obteve com 10 vezes mais tempo que o exemplo mais demorado obtido pela heurística.

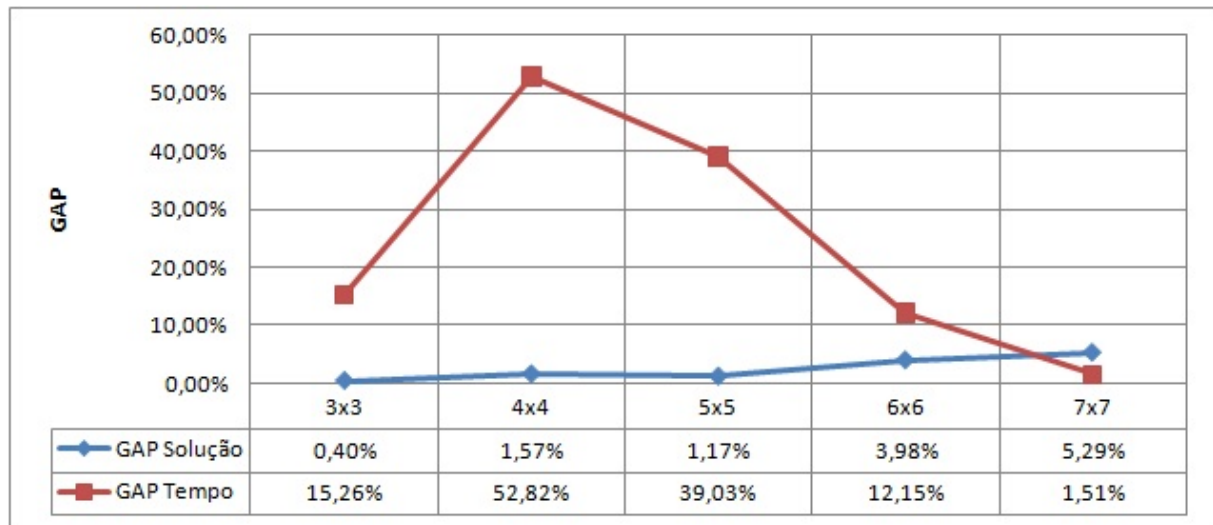


Figura 4.1: Comparativos dos resultados 3x3 a 7x7

Fonte: O autor

No caso dos problemas 8x8, o tempo médio de processamento foi de 3606,23ms (todos com 5000 iterações), com desvio padrão de 266,21ms, e o problema mais demorado foi com o tempo de 4193ms. O intervalo de confiança de 95% para o tempo médio é (3510,97 ; 3701,50). Assim o modelo matemático teve seu tempo de execução limitado em 42s (baseado no maior tempo entre todos os testes). Os resultados obtidos em cada exemplo estão na Tabela B.6 (página 63).

Os valores negativos para o $GAP_{\text{solução}}$ indicam que a solução encontrada pela heurística com 10% do tempo para o PIM é menor que a solução obtida pelo modelo matemático, por exemplo, para o problema 8x8_Ex01, $GAP_{\text{solução}} = -4,17\%$ mostra que a solução obtida pela heurística é 4,17% menor que a resposta do modelo matemático. Se o valor for positivo, mostra quanto por cento a resposta da heurística é maior que a resposta do modelo matemático, que foi interrompido no tempo estipulado.

A média dos $GAP_{\text{solução}}$ para os problemas 8x8 é de $-3,80\%$, ou seja, na média, as soluções obtidas pela heurística com 10% do tempo de execução do PIM é 3,80% melhor.

Pode-se observar o comportamento das soluções na Figura 4.2 (página 41), onde a linha em azul com a legenda Resultado Heurística apresenta o melhor valor obtido para o *Makespan* pela heurística e a linha em vermelho com a legenda Resultado PIM, para o

modelo matemático, com o tempo especificado acima de 42s.

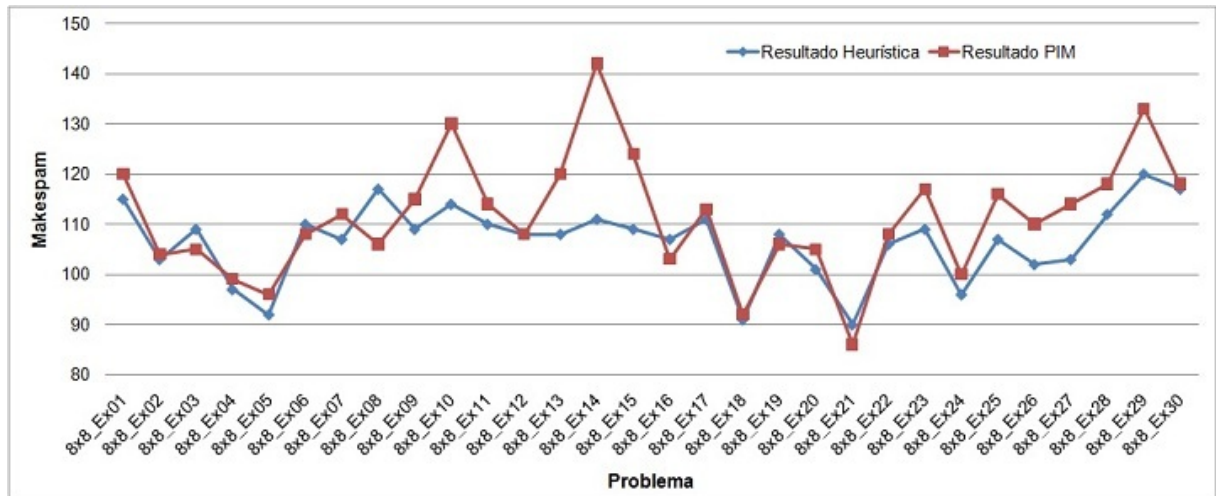


Figura 4.2: Resultados 8x8

Fonte: O autor

Foi feita a mesma construção e análise para os problemas 9x9. O tempo médio de execução da heurística nos problemas 9x9 foi de 4744,90ms (também com 5000 iterações), com desvio padrão de 56,08ms, e o problema mais demorado teve o tempo de 4859ms. O intervalo de confiança de 95% para o tempo médio é (4724,83 ; 4764,97). Assim, o tempo de processamento do modelo matemático foi limitado em 49s (novamente baseado no maior tempo de processamento).

Os resultados obtidos para cada problema encontram-se na Tabela B.7 (página 64). A média dos $GAP_{\text{solução}}$ para os problemas 9x9 é de $-24,92\%$, ou seja, na média, as soluções obtidas pela heurística com 10% do tempo de execução do PIM é 24,92% melhor.

A Figura 4.3 (página 42) apresenta os resultados obtidos pela heurística e pelo modelo matemático, valendo os mesmos comentários do caso 8x8. Nota-se que em alguns problemas, como o 9x9_Ex04 não se tem um resultado para o modelo matemático, isto se deve ao fato que o mesmo não encontrou uma primeira solução factível para então começar a otimizar o problema. Assim, o modelo matemático começou a ficar inviável para tempos pequenos (na ordem de segundos ou poucos minutos) mesmo para obter a primeira solução factível.

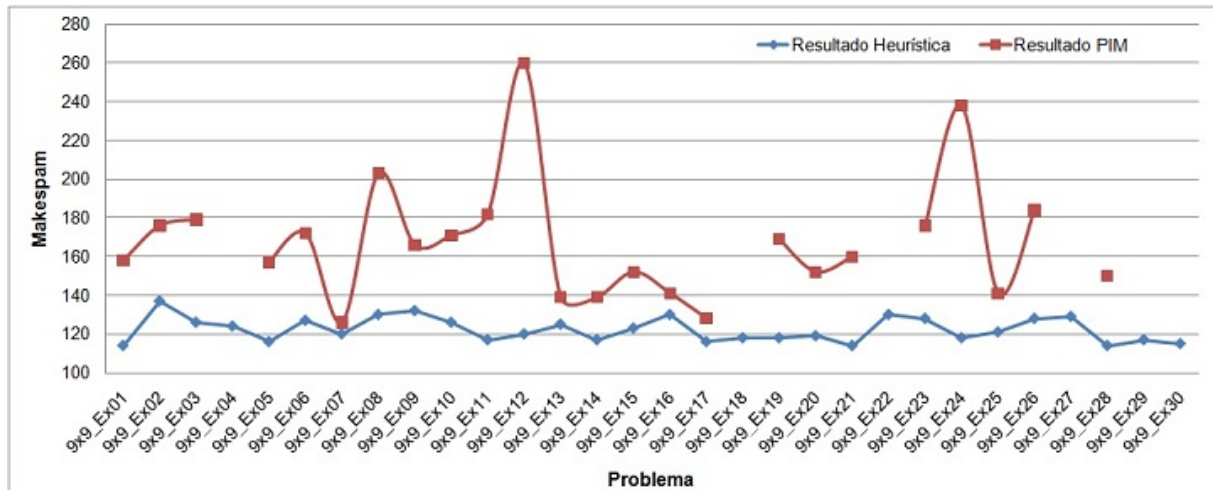


Figura 4.3: Resultados 9x9

Fonte: O autor

4.3 Problemas 10x10, 15x15 e 20x20

Passando para os problemas da ordem 10x10, inicialmente foram realizados os mesmos tipos de teste que os problemas 8x8 e 9x9, porém, mesmo aumentando o tempo limite de processamento do modelo matemático em até 15 minutos, sendo que o maior tempo que a heurística levou para esta ordem foi de 6912 ms, o mesmo não encontrava uma solução inicial factível.

Neste ponto, foi pensando uma outra abordagem ao problema de comparar respostas: utilizar a solução obtida pela heurística, que já é uma solução otimizada e factível, como solução inicial para o modelo matemático. Então foi procedido desta forma e sem limitar o tempo de execução do modelo exato.

Porém, o mesmo não conseguiu atingir o ótimo, sendo que em um exemplo 20x20 permaneceu mais de 96 horas em execução e o ótimo não foi encontrado. De forma semelhante foi com os exemplos 10x10 e 15x15. Este fato era algo esperado, pois para esta ordem de problemas, o número de restrições e variáveis binárias do modelo matemático o tornam de impossível solução, por pertencerem à classe dos problemas *NP-Hard*.

Então foram passados ao modelo exato as soluções obtidas pela heurística, mas limitando-se o tempo de execução em 1 minuto e em 2 minutos. Com valores de tempo

maiores, ocorreu o mesmo tipo de problema de estacionar em uma solução e não atingir o ótimo ou de extrapolar a memória do computador.

Para os problemas da ordem 10x10, 15x15 e 20x20, os resultados obtidos estão nas Tabelas B.8 (página 65), B.9 (página 66) e B.10 (página 67), respectivamente. Em todas as ordens, os problemas foram executados com 5000 iterações na heurística. A Tabela 4.4 (página 43) apresenta um resumo dos resultados sobre os problemas de ordem 10x10, 15x15 e 20x20.

Tabela 4.4: Comparativos resultados 10x10, 15x15 e 20x20

Fonte: O autor

Tamanho	Média Tempo Heurística (ms)	Percentual de Melhoria	
		1min.	2min.
10x10	6535,17	4,63%	5,39%
15x15	22682,13	0,24%	4,46%
20x20	58575,63	0,02%	1,26%

Os problemas da ordem 10x10 continuam com tempos de processamento próximos de 6500ms, os problemas da ordem 15x15 (considerados grande na literatura) passaram para 22682ms e mesmo para problemas da ordem 20x20, a heurística não leva mais que um minuto (em média). Os desvios padrões são, 95,07 ms, 206,63 ms e 661,12 ms respectivamente. Assim, os intervalos de confiança de 95% para o tempo de processamento (em ms) são: (6501,14 ; 6569,19), (22608,19 ; 22765,08) e (58339,06 ; 58812,21).

Na Figura 4.4 (página 44), pode-se observar que após um minuto no modelo matemático, houve melhorias em todas as soluções, de todos os exemplos da ordem 10x10, sendo a maior 8,07%. Após dois minutos, o ganho sobre um minuto foi menor, mas ainda houve soluções melhoradas, onde a maior de todas as melhorias sobre a solução inicial foi de 10,53%.

Nos problemas da ordem 15x15, após um minuto houve melhoras em 7 problemas, com a melhor melhoria de apenas 1,39%. Após dois minutos, houve melhoras em todos os problemas, conforme ilustra a Figura 4.5 (página 44), sendo a melhor melhoria de 9,38% sobre a solução inicial.

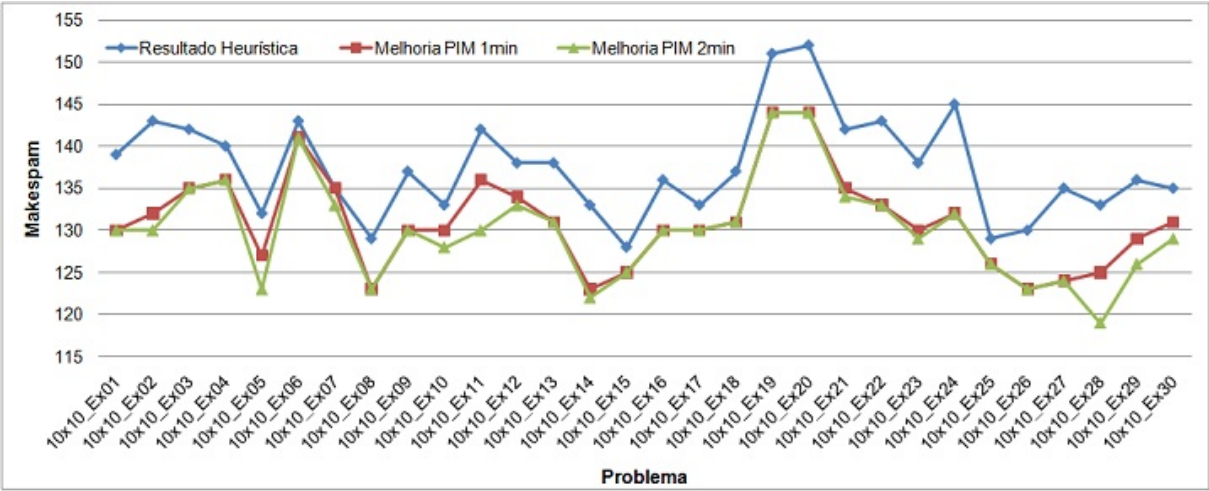


Figura 4.4: Resultados 10x10

Fonte: O autor

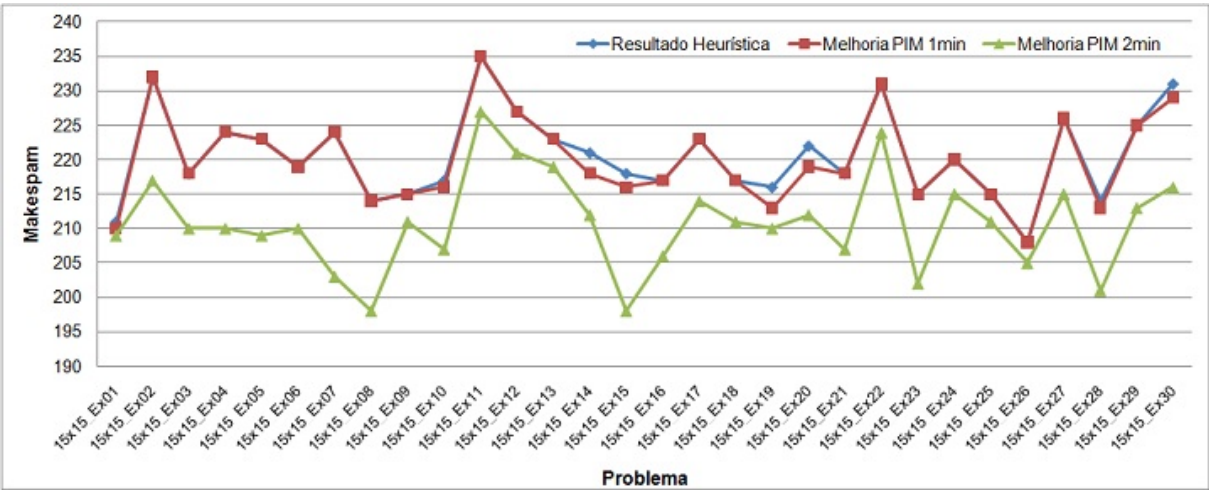


Figura 4.5: Resultados 15x15

Fonte: O autor

Finalmente, para os problemas da ordem 20x20, após um minuto houve melhoria em apenas um problema (20x20.Ex12), e esta foi de apenas 0,65%. Limitado a dois minutos, o modelo exato obteve como melhor melhoria 3,68%, e houve melhorias em 20 dos 30 problemas.

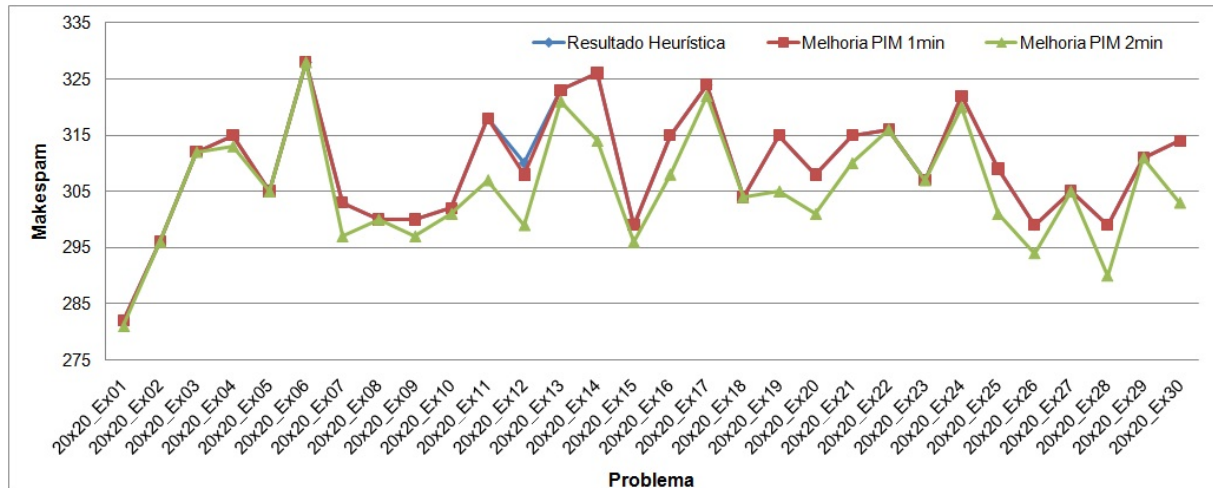


Figura 4.6: Resultados 20x20

Fonte: O autor

Ao final de todos os testes, ainda foi comparado o tempo médio que cada iteração levou para todas as ordem de problema testados. A Tabela 4.5 (página 46) apresenta os resultados.

Tabela 4.5: Tempos médios das Iterações (ms) 3x3 a 20x20

Fonte: O autor

Tamanho	Tempos médios	Iterações	Tempo/Iteração
3x3	5,93	100	0,06
4x4	59,90	500	0,12
5x5	215,53	1000	0,22
6x6	1718,63	5000	0,34
7x7	2688,57	5000	0,54
8x8	3606,23	5000	0,72
9x9	4744,90	5000	0,95
10x10	6535,17	5000	1,31
15x15	22682,13	5000	4,54
20x20	58575,63	5000	11,72

Conclusões

Ao final desta tese, conclui-se que todos os objetivos foram devidamente completos. A revisão da literatura foi realizada e nada semelhante foi encontrado para resolver o problema abordado por este trabalho. Após estudar o algoritmo de Giffler e Thompson (1960), foi proposto algoritmo heurístico descrito, como forma de otimizar o *makespan* no ambiente de oficina de máquinas, com tempo de *setup* variável, dependente da sequência.

O algoritmo foi implementado, na linguagem Visual Basic .net e ficou comprovado que o tempo de processamento é rápido, mesmo para os problemas grandes, cada iteração dura uma fração de segundo. E não são necessárias muitas iterações para obter uma solução otimizada. Como o processo de cada iteração do algoritmo apresentado é independente da iteração anterior, não há garantias em qual iteração a melhor solução é obtida. Quanto mais iterações foi dado ao algoritmo, apenas há mais possibilidades de soluções para se procurar, em função da aleatoriedade do Passo 4. Assim sendo, não houve a necessidade do número de iterações ser muito grande (1000000 por exemplo).

O mesmo foi aplicado a uma série de testes, de diversos tamanhos, e a escolha da comparação com o resultado exato sobre um modelo de programação linear mista foi por não ter sido encontrada uma base de dados com problemas resolvidos de forma heurística. Como citado na revisão bibliográfica, o problema de sequenciamento contempla muitas possibilidades de problemas, no sentido de decisões, ambientes, objetivos, características inerentes a um estudo de caso, etc. Também tal escolha foi para comparar com a melhor solução possível, a exata, até onde era possível de fazê-lo.

E surgiu como um novo objetivo durante a realização dos testes, a ideia de utilizar a solução otimizada encontrada pela heurística, como solução inicial factível para o modelo exato, diante da dificuldade do modelo encontrar esta primeira resposta. E esta ideia mostrou-se eficiente, pois em diversos testes, nos problemas de grande porte, mesmo após muitas horas de processamento, o modelo exato não encontrou a primeira solução factível.

Enquanto que uma solução otimizada pela heurística não leva mais que um minuto para ser obtida, e com mais dois minutos no modelo exato, esta solução é melhorada. Este modelo híbrido heurístico-exato pode fornecer soluções boas para problemas reais.

Como sugestão para trabalhos futuros:

- O estudo de outros critérios para seleção da tarefa a ser sequenciada, além do tempo de processamento da mesma, como o prazo de entrega ou o custo do processamento da tarefa.
- A criação de critérios de parada para o processo iterativo sobre a heurística, além do número total de iterações. Por exemplo, para o processo iterativo após k iterações sem melhorar o resultado, o que tornaria mais rápido ainda a obtenção de soluções otimizadas.
- Implementar uma estrutura que permita salvar as diferentes respostas obtidas pela heurística com um mesmo melhor valor de *makespan*.
- Fazer testes para problemas que são casos particulares do ambiente de oficina de máquinas, como máquinas em série.
- Utilizar o resultado obtido pela heurística como ponto factível inicial para outros modelos exatos que não exija que todas as tarefas passem por todas as máquinas uma e apenas uma vez, pois a heurística permite a resolução de mais tipos de problemas.
- Inclusão de um algoritmo de busca local no **Passo 2** do algoritmo proposto nesta tese.

Referências Bibliográficas

- [1] ALVAREZ-VALDES, R.; FUERTE, A.; TAMARIT, J. M.; GIMENEZ, G.; RAMOS, R. *A heuristic to schedule flexible job-shop in a glass factory*. European Journal of Operational Research, 165, 525-534. 2005.
- [2] CHEUNG, W.; ZHOU, H. *Using Genetic Algorithms and Heuristics for Job Shop Scheduling with Sequence-Dependent Setup Times*. Annals of Operations Research, 107, 65-81, 2001.
- [3] COLIN, E. C. *Pesquisa operacional : 170 aplicações em estratégia, finanças, logística, produção, marketing e vendas*. Rio de Janeiro: LTC, 2007. xix, 501 p.
- [4] GIFFLER, J.; THOMPSON, G. L. *Algorithms for solving production scheduling problems*. Operations Research, 8(4), 487-503. 1960.
- [5] GEORGIADIS, P.; MICHALOUDIS, C. *Real-time production planning and control system for job-shop manufacturing: A system dynamics analysis*. European Journal of Operational Research, 216, 94-104. 2012.
- [6] HUANG, K.-L.; LIAO, C.-J. *Ant colony optimization combined with taboo search for the job shop scheduling problem*. Computers and Operational Research, 35, 1030-1046. 2008.
- [7] IERAPETRITOU, M. G.; FLOUDAS, C. A. *Effective continuous-time formulation for short-term scheduling: 2. Continuous and semicontinuous processes*. Industrial Engineering and Chemical Research, 37, 4360-4374. 1998.
- [8] IERAPETRITOU, M. G.; HENE, T. S.; FLOUDAS, C. A. *Short-term scheduling of batch plants with multiple intermediate due dates*. Computers and Chemical Engineering, 23, 515-518. 1999.

- [9] KARIMI-NASAB, M.; SEYEDHOSEINI, S. M. *Multi-level lot sizing and job shop scheduling with compressible process times: A cutting plane approach*. European Journal of Operational Research, 231, 598-616. 2013.
- [10] KIM, J. U.; KIM, Y. D. *Simulated annealing and genetic algorithms for scheduling products with multi-level product structure*. Computers & Operations Research, 23, 857-868. 1996.
- [11] LIU, T.-K.; TSAI, J.-T.; CHOU, J.-H. *Improved genetic algorithm for the job-shop scheduling problems*. International Journal of Advanced Manufacturing Technology, 27, 1021-1029. 2006.
- [12] LUO, H.; HUANG, G. Q.; ZHANG, Y.; DAI Q.; CHEN, X. *Two-stage hybrid batching flowshop scheduling with blocking and machine availability constraints using genetic algorithm*. Robotic and Computer-Integrated Manufacturing, 25, 962-971. 2009.
- [13] LUSTOSA L.; MESQUITA, M. A.; QUELHAS, O.; OLIVEIRA, R. *Planejamento e Controle da Produção*. Rio de Janeiro: Editora Elsevier, 2008. xiv, 357 p.
- [14] MANIKAS, A.; CHANG, Y.-L. *Multi-criteria sequence-dependent job shop scheduling using genetic algorithms*. Computer & Industrial Engineering, 56, 179-185. 2009.
- [15] MATI, Y.; DAUZÈRE-PÉRÈS, S.; LAHLOU, C. *A general approach for optimizing regular criteria in the job-shop scheduling problem*. European Journal of Operational Research, 212, 33-42. 2011.
- [16] MONKMAN, S. K.; MORRICE, D. J.; BARD, J. F. *A production scheduling heuristic for an electronics manufacturer with sequence-dependent setup costs*. European Journal of Operational Research, 187, 1100-1114. 2008.
- [17] NOWICKI, E.; SMUTNICKI, C. *An advanced tabu search algorithm for the job shop problem*. Journal of Scheduling 8, 145-159. 2005.
- [18] OZTURK, C.; ORNECK, A. M. *Capacitated lot sizing with linked lots for general products structures in job shops*. Computers & Industrial Engineering, 58, 151-164. 2010.

- [19] PARTHANADEE, P.; BUDDHAKULSOMSIRI, J. *Simulation modeling and analysis for production scheduling using real-time dispatching rules: A case study in canned fruit industry*. Computers and Electronics in Agriculture, 70, 245-255. 2010.
- [20] PROUD, J. F. *Master scheduling : practical guide to competitive manufacturing*. 2nd ed. New York : J. Wiley & Sons, c1999. 610 p.
- [21] QIAN, B.; WANG, L.; HUANG, D.; WANG, X. *Scheduling multi-objective job shops using a memetic algorithm based on differential evolution*. International Journal of Advanced Manufacturing Technology, 35, 1014-1027. 2008.
- [22] ROCHA, P.L.; RAVETTI, M. G.; MATEUS, G. R. *The metaheuristic GRASP as an upper bound for a branch and bound algorithm in a scheduling problem with non-parallel machines and sequence-dependent setup times*. Proceedings of 4th EU/ME workshop: Design and evaluation of advanced hybrid meta-heuristics, 62-67. Nottingham, UK. 2004.
- [23] RUIZ, R.; MAROTO, C.; ALCARAZ, J. *Solving the flowshop scheduling problem with sequence dependent setup times using advanced metaheuristics*. European Journal of Operational Research, 165, 34-54. 2005.
- [24] SADEGHEIH, A. *Scheduling problem using genetic algorithm, simulated annealing and the effects of parameter values on GA performance*. Applied Mathematical Modelling, 30, 147-154. 2006.
- [25] SELS, V.; CRAEYMEERSCH, K.; VANHOUCKE, M. *A hybrid single and dual population search procedure for the job shop scheduling problem*. European Journal of Operational Research, 215, 512-523. 2011.
- [26] SHEN, L.; BUSCHER, U. *Solving the serial batching problem in job shop manufacturing systems*. European Journal of Operational Research, 221, 14-26. 2012.
- [27] SCHWINDT, C.; TRAUTMANN, N. *Batch scheduling in process industries: An application of resource-constrained project scheduling*. OR Spectrum, 22, 501-524, 2000.
- [28] TASGETIREN, M. F.; LIANG, Y. C.; SEVKLI, M.; GENCYILMAZ, G. *A particle swarm optimization algorithm for makespan and total flowtime minimization*

- in the permutation flowshop sequencing problem*. European Journal of Operational Research, 177, 1930-1947. 2007.
- [29] WONG, T. C.; CHAN, F. T. S.; CHAN, L. Y. *A resource-constrained assembly job shop scheduling problem with lot streaming technique*. Computers & Industrial Engineering, 57, 983-995.2009.
- [30] ZHOU, H.; CHEUNG, W.; LEUNG, L. *Minimizing weighted tardiness of job-shop scheduling using a hybrid genetic algorithm*. European Journal of Research, 194 (1), 637-649. 2009.

Modelo Matemático

O arquivo *.lp* com o modelo matemático (PIM) para o problema 3x3_Ex01, e então o arquivo *.sol* com a solução do problema no padrão para leitura do Cplex, onde não são necessárias as variáveis nulas.

Modelo exato:

\Problema: Modelo Exato 3x3_Ex01.lp

Minimize

obj: Cmax

Subject to

c1:t_1_1 - t_2_1 <= -3

c2:t_2_2 - t_1_2 <= -1

c3:t_3_3 - t_2_3 <= -2

c4:t_2_1 - t_3_1 <= -2

c5:t_1_2 - t_3_2 <= -4

c6:t_2_3 - t_1_3 <= -4

c7:t_3_1 - Cmax <= - 2

c8:t_3_2 - Cmax <= - 3

c9:t_1_3 - Cmax <= - 1

c10:t_1_1 + 1x_1_1_2 - t_1_2 + 1000x_1_1_2 <= 997

c11:t_1_1 + 2x_1_1_3 - t_1_3 + 1000x_1_1_3 <= 997

c12:t_1_2 + 2x_1_2_1 - t_1_1 + 1000x_1_2_1 <= 996

c13:t_1_2 + 3x_1_2_3 - t_1_3 + 1000x_1_2_3 <= 996

c14:t_1_3 + 1x_1_3_1 - t_1_1 + 1000x_1_3_1 <= 999

c15:t_1_3 + 4x_1_3_2 - t_1_2 + 1000x_1_3_2 <= 999

c16:t_2_1 + 2x_2_1_2 - t_2_2 + 1000x_2_1_2 <= 998

$$\begin{aligned}
c17: & t_{2_1} + 1x_{2_1_3} - t_{2_3} + 1000x_{2_1_3} \leq 998 \\
c18: & t_{2_2} + 1x_{2_2_1} - t_{2_1} + 1000x_{2_2_1} \leq 999 \\
c19: & t_{2_2} + 4x_{2_2_3} - t_{2_3} + 1000x_{2_2_3} \leq 999 \\
c20: & t_{2_3} + 3x_{2_3_1} - t_{2_1} + 1000x_{2_3_1} \leq 996 \\
c21: & t_{2_3} + 1x_{2_3_2} - t_{2_2} + 1000x_{2_3_2} \leq 996 \\
c22: & t_{3_1} + 3x_{3_1_2} - t_{3_2} + 1000x_{3_1_2} \leq 998 \\
c23: & t_{3_1} + 2x_{3_1_3} - t_{3_3} + 1000x_{3_1_3} \leq 998 \\
c24: & t_{3_2} + 3x_{3_2_1} - t_{3_1} + 1000x_{3_2_1} \leq 997 \\
c25: & t_{3_2} + 1x_{3_2_3} - t_{3_3} + 1000x_{3_2_3} \leq 997 \\
c26: & t_{3_3} + 3x_{3_3_1} - t_{3_1} + 1000x_{3_3_1} \leq 998 \\
c27: & t_{3_3} + 2x_{3_3_2} - t_{3_2} + 1000x_{3_3_2} \leq 998 \\
c28: & + x_{1_0_1} + x_{1_0_2} + x_{1_0_3} = 1 \\
c29: & + x_{1_1_0} + x_{1_1_2} + x_{1_1_3} = 1 \\
c30: & + x_{1_2_0} + x_{1_2_1} + x_{1_2_3} = 1 \\
c31: & + x_{1_3_0} + x_{1_3_1} + x_{1_3_2} = 1 \\
c32: & + x_{2_0_1} + x_{2_0_2} + x_{2_0_3} = 1 \\
c33: & + x_{2_1_0} + x_{2_1_2} + x_{2_1_3} = 1 \\
c34: & + x_{2_2_0} + x_{2_2_1} + x_{2_2_3} = 1 \\
c35: & + x_{2_3_0} + x_{2_3_1} + x_{2_3_2} = 1 \\
c36: & + x_{3_0_1} + x_{3_0_2} + x_{3_0_3} = 1 \\
c37: & + x_{3_1_0} + x_{3_1_2} + x_{3_1_3} = 1 \\
c38: & + x_{3_2_0} + x_{3_2_1} + x_{3_2_3} = 1 \\
c39: & + x_{3_3_0} + x_{3_3_1} + x_{3_3_2} = 1 \\
c40: & + x_{1_1_0} + x_{1_2_0} + x_{1_3_0} = 1 \\
c41: & + x_{1_0_1} + x_{1_2_1} + x_{1_3_1} = 1 \\
c42: & + x_{1_0_2} + x_{1_1_2} + x_{1_3_2} = 1 \\
c43: & + x_{1_0_3} + x_{1_1_3} + x_{1_2_3} = 1 \\
c44: & + x_{2_1_0} + x_{2_2_0} + x_{2_3_0} = 1 \\
c45: & + x_{2_0_1} + x_{2_2_1} + x_{2_3_1} = 1 \\
c46: & + x_{2_0_2} + x_{2_1_2} + x_{2_3_2} = 1 \\
c47: & + x_{2_0_3} + x_{2_1_3} + x_{2_2_3} = 1 \\
c48: & + x_{3_1_0} + x_{3_2_0} + x_{3_3_0} = 1
\end{aligned}$$

$$c49: + x_{3_0_1} + x_{3_2_1} + x_{3_3_1} = 1$$

$$c50: + x_{3_0_2} + x_{3_1_2} + x_{3_3_2} = 1$$

$$c51: + x_{3_0_3} + x_{3_1_3} + x_{3_2_3} = 1$$

$$c52: + x_{1_0_1} + x_{1_1_0} \leq 1$$

$$c53: + x_{1_0_2} + x_{1_2_0} \leq 1$$

$$c54: + x_{1_0_3} + x_{1_3_0} \leq 1$$

$$c55: + x_{1_1_2} + x_{1_2_1} \leq 1$$

$$c56: + x_{1_1_3} + x_{1_3_1} \leq 1$$

$$c57: + x_{1_2_3} + x_{1_3_2} \leq 1$$

$$c58: + x_{2_0_1} + x_{2_1_0} \leq 1$$

$$c59: + x_{2_0_2} + x_{2_2_0} \leq 1$$

$$c60: + x_{2_0_3} + x_{2_3_0} \leq 1$$

$$c61: + x_{2_1_2} + x_{2_2_1} \leq 1$$

$$c62: + x_{2_1_3} + x_{2_3_1} \leq 1$$

$$c63: + x_{2_2_3} + x_{2_3_2} \leq 1$$

$$c64: + x_{3_0_1} + x_{3_1_0} \leq 1$$

$$c65: + x_{3_0_2} + x_{3_2_0} \leq 1$$

$$c66: + x_{3_0_3} + x_{3_3_0} \leq 1$$

$$c67: + x_{3_1_2} + x_{3_2_1} \leq 1$$

$$c68: + x_{3_1_3} + x_{3_3_1} \leq 1$$

$$c69: + x_{3_2_3} + x_{3_3_2} \leq 1$$

Binaries

$x_{1_1_0}$ $x_{1_2_0}$ $x_{1_3_0}$

$x_{1_0_1}$ $x_{1_2_1}$ $x_{1_3_1}$

$x_{1_0_2}$ $x_{1_1_2}$ $x_{1_3_2}$

$x_{1_0_3}$ $x_{1_1_3}$ $x_{1_2_3}$

$x_{2_1_0}$ $x_{2_2_0}$ $x_{2_3_0}$

$x_{2_0_1}$ $x_{2_2_1}$ $x_{2_3_1}$

$x_{2_0_2}$ $x_{2_1_2}$ $x_{2_3_2}$

$x_{2_0_3}$ $x_{2_1_3}$ $x_{2_2_3}$

$x_{3_1_0}$ $x_{3_2_0}$ $x_{3_3_0}$

$x_{3_0_1}$ $x_{3_2_1}$ $x_{3_3_1}$

```

x_3_0_2 x_3_1_2 x_3_3_2
x_3_0_3 x_3_1_3 x_3_2_3
End

```

Arquivo de solução:

```

<?xml version = "1.0" encoding="UTF-8" standalone="yes"?>
<CPLEXSolution version="1.2">
  <variables>
    <variable name="Cmax" index="0" value="13"/>
    <variable name="t_2_2" index="1" value="0"/>
    <variable name="t_3_3" index="2" value="0"/>
    <variable name="t_1_1" index="3" value="0"/>
    <variable name="t_2_1" index="4" value="3"/>
    <variable name="t_3_1" index="5" value="5"/>
    <variable name="t_1_2" index="6" value="4"/>
    <variable name="t_2_3" index="7" value="6"/>
    <variable name="t_3_2" index="8" value="10"/>
    <variable name="t_1_3" index="9" value="11"/>
    <variable name="x_2_0_2" index="10" value="1"/>
    <variable name="x_3_0_3" index="11" value="1"/>
    <variable name="x_1_0_1" index="12" value="1"/>
    <variable name="x_2_2_1" index="13" value="1"/>
    <variable name="x_3_3_1" index="14" value="1"/>
    <variable name="x_1_1_2" index="15" value="1"/>
    <variable name="x_2_1_3" index="16" value="1"/>
    <variable name="x_3_1_2" index="17" value="1"/>
    <variable name="x_1_2_3" index="18" value="1"/>
    <variable name="x_1_3_0" index="19" value="1"/>
    <variable name="x_2_3_0" index="20" value="1"/>
    <variable name="x_3_2_0" index="21" value="1"/>
  </variables>
</CPLEXSolution>

```

Resumo dos resultados

Tabela B.1: Resultados Testes 3x3

Fonte: O autor

Teste n°	Problema	Resultado Heurística	Tempo (ms)	Resultado PIM	Tempo (ms)	GAP Solução	GAP Tempo
1	3x3_Ex01	13	6	13	30	0,00%	20%
2	3x3_Ex02	38	8	38	230	0,00%	3,48%
3	3x3_Ex03	37	5	37	50	0,00%	10,00%
4	3x3_Ex04	26	7	26	50	0,00%	14,00%
5	3x3_Ex05	24	5	24	30	0,00%	16,67%
6	3x3_Ex06	35	7	35	250	0,00%	2,80%
7	3x3_Ex07	33	6	33	50	0,00%	12,00%
8	3x3_Ex08	37	5	37	50	0,00%	10,00%
9	3x3_Ex09	36	6	36	30	0,00%	20,00%
10	3x3_Ex10	27	6	27	30	0,00%	20,00%
11	3x3_Ex11	38	5	38	50	0,00%	10,00%
12	3x3_Ex12	25	7	25	30	0,00%	23,33%
13	3x3_Ex13	34	9	34	10	0,00%	90,00%
14	3x3_Ex14	28	5	28	30	0,00%	16,67%
15	3x3_Ex15	33	7	33	60	0,00%	11,67%
16	3x3_Ex16	31	5	31	50	0,00%	10,00%
17	3x3_Ex17	28	8	28	60	0,00%	13,33%
18	3x3_Ex18	30	6	30	110	0,00%	5,45%
19	3x3_Ex19	32	5	32	50	0,00%	10,00%
20	3x3_Ex20	31	5	31	30	0,00%	16,67%
21	3x3_Ex21	36	5	34	30	5,88%	16,67%
22	3x3_Ex22	35	5	34	110	2,94%	4,54%
23	3x3_Ex23	34	5	34	50	0,00%	10,00%
24	3x3_Ex24	26	6	26	30	0,00%	20,00%

(continua na próxima página)

Tabela B.1: Resultados Testes 3x3 - continuação

Teste n°	Problema	Resultado Heurística	Tempo (ms)	Resultado PIM	Tempo (ms)	GAP Solução	GAP Tempo
25	3x3_Ex25	30	5	30	160	0,00%	3,13%
26	3x3_Ex26	32	7	31	30	3,23%	23,33%
27	3x3_Ex27	39	7	39	50	0,00%	14,00%
28	3x3_Ex28	25	5	25	50	0,00%	10,00%
29	3x3_Ex29	33	5	33	50	0,00%	10,00%
30	3x3_Ex30	28	5	28	50	0,00%	10,00%

Tabela B.2: Resultados Testes 4x4

Fonte: O autor

Teste n°	Problema	Resultado Heurística	Tempo (ms)	Resultado PIM	Tempo (ms)	GAP Solução	GAP Tempo
1	4x4.Ex01	52	62	50	530	4,00%	11,70%
2	4x4.Ex02	46	79	46	90	0,00%	87,78%
3	4x4.Ex03	38	80	38	140	0,00%	57,14%
4	4x4.Ex04	48	55	48	200	0,00%	27,50%
5	4x4.Ex05	56	55	56	90	0,00%	61,11%
6	4x4.Ex06	43	75	43	90	0,00%	83,33%
7	4x4.Ex07	46	55	46	110	0,00%	50,00%
8	4x4.Ex08	44	56	44	140	0,00%	40,00%
9	4x4.Ex09	47	56	44	110	6,82%	50,91%
10	4x4.Ex10	45	55	45	90	0,00%	61,11%
11	4x4.Ex11	42	54	42	160	0,00%	33,75%
12	4x4.Ex12	52	55	52	110	0,00%	50,00%
13	4x4.Ex13	37	78	37	140	0,00%	55,71%
14	4x4.Ex14	51	57	48	110	6,25%	51,82%
15	4x4.Ex15	46	59	46	140	0,00%	42,14%
16	4x4.Ex16	53	55	52	90	1,92%	61,11%
17	4x4.Ex17	53	57	50	60	6,00%	95,00%
18	4x4.Ex18	56	58	56	130	0,00%	44,62%
19	4x4.Ex19	43	60	43	160	0,00%	37,50%
20	4x4.Ex20	44	79	44	170	0,00%	46,47%
21	4x4.Ex21	49	56	47	160	4,26%	35,00%
22	4x4.Ex22	44	54	41	80	7,32%	6,75%
23	4x4.Ex23	45	55	44	80	2,27%	68,75%
24	4x4.Ex24	43	57	43	110	0,00%	51,82%
25	4x4.Ex25	54	58	54	140	0,00%	41,43%
26	4x4.Ex26	49	54	48	120	2,08%	45,00%
27	4x4.Ex27	52	53	49	90	6,12%	58,89%
28	4x4.Ex28	45	56	45	60	0,00%	93,33%
29	4x4.Ex29	43	58	43	170	0,00%	34,12%
30	4x4.Ex30	42	56	42	140	0,00%	40,00%

Tabela B.3: Resultados Testes 5x5

Fonte: O autor

Teste n°	Problema	Resultado Heurística	Tempo (ms)	Resultado PIM	Tempo (ms)	GAP Solução	GAP Tempo
1	5x5_Ex01	59	189	58	267	1,72%	70,79%
2	5x5_Ex02	58	190	58	500	0,00%	38,00%
3	5x5_Ex03	60	195	60	640	0,00%	30,47%
4	5x5_Ex04	55	193	54	370	1,85%	52,16%
5	5x5_Ex05	50	193	49	360	2,04%	53,61%
6	5x5_Ex06	58	254	58	330	0,00%	76,97%
7	5x5_Ex07	57	298	57	1720	0,00%	17,33%
8	5x5_Ex08	53	196	53	360	0,00%	54,44%
9	5x5_Ex09	52	276	48	410	8,33%	67,32%
10	5x5_Ex10	64	191	64	1670	0,00%	11,44%
11	5x5_Ex11	58	253	58	1360	0,00%	18,60%
12	5x5_Ex12	53	198	53	280	0,00%	70,71%
13	5x5_Ex13	67	192	67	640	0,00%	30,00%
14	5x5_Ex14	54	272	54	2450	0,00%	11,10%
15	5x5_Ex15	57	190	55	1250	3,64%	15,20%
16	5x5_Ex16	58	191	58	360	0,00%	53,06%
17	5x5_Ex17	59	220	59	470	0,00%	46,81%
18	5x5_Ex18	51	198	51	390	0,00%	50,77%
19	5x5_Ex19	62	256	62	1650	0,00%	15,52%
20	5x5_Ex20	59	233	57	1390	3,51%	16,76%
21	5x5_Ex21	65	266	61	620	6,56%	42,90%
22	5x5_Ex22	59	193	57	720	3,51%	26,81%
23	5x5_Ex23	64	192	64	1760	0,00%	10,91%
24	5x5_Ex24	53	193	52	230	1,92%	83,91%
25	5x5_Ex25	58	193	58	1360	0,00%	14,19%
26	5x5_Ex26	57	198	57	380	0,00%	52,11%
27	5x5_Ex27	55	200	55	1110	0,00%	18,02%
28	5x5_Ex28	53	196	52	2070	1,92%	9,47%
29	5x5_Ex29	58	192	58	410	0,00%	46,83%
30	5x5_Ex30	56	265	56	410	0,00%	64,63%

Tabela B.4: Resultados Testes 6x6

Fonte: O autor

Teste n°	Problema	Resultado Heurística	Tempo (ms)	Resultado PIM	Tempo (ms)	GAP Solução	GAP Tempo
1	6x6_Ex01	72	1573	68	10870	5,88%	14,47%
2	6x6_Ex02	70	1943	67	43590	4,48%	4,46%
3	6x6_Ex03	63	1645	62	7710	1,61%	21,34%
4	6x6_Ex04	83	1962	78	14290	6,41%	13,73%
5	6x6_Ex05	84	1887	79	27500	6,33%	6,86%
6	6x6_Ex06	72	1576	70	23030	2,86%	6,84%
7	6x6_Ex07	83	1604	81	97250	2,47%	1,65%
8	6x6_Ex08	78	2057	76	34350	2,63%	5,99%
9	6x6_Ex09	74	1791	73	8640	1,37%	20,73%
10	6x6_Ex10	78	1602	74	47410	5,41%	3,38%
11	6x6_Ex11	74	1831	71	13760	4,23%	13,31%
12	6x6_Ex12	80	1677	78	73200	2,56%	2,29%
13	6x6_Ex13	79	1692	76	12350	3,95%	13,70%
14	6x6_Ex14	73	1634	72	13060	1,39%	12,51%
15	6x6_Ex15	83	1584	80	30560	3,75%	5,18%
16	6x6_Ex16	73	1709	68	29300	7,35%	5,83%
17	6x6_Ex17	76	1587	72	8220	5,56%	19,31%
18	6x6_Ex18	72	1662	67	10060	7,46%	16,52%
19	6x6_Ex19	72	1564	69	14090	4,35%	11,10%
20	6x6_Ex20	73	2163	71	8110	2,82%	26,67%
21	6x6_Ex21	82	1599	81	28360	1,23%	5,64%
22	6x6_Ex22	74	1848	72	10980	2,78%	16,83%
23	6x6_Ex23	67	1924	65	7330	3,08%	26,25%
24	6x6_Ex24	79	1729	76	74490	3,95%	2,32%
25	6x6_Ex25	76	1612	70	11870	8,57%	13,58%
26	6x6_Ex26	70	1577	67	4130	4,48%	38,18%
27	6x6_Ex27	73	1589	71	18500	2,82%	8,59%
28	6x6_Ex28	76	1587	74	248810	2,70%	0,64%
29	6x6_Ex29	76	1696	73	37470	4,11%	4,53%
30	6x6_Ex30	73	1655	71	7470	2,82%	22,16%

Tabela B.5: Resultados Testes 7x7

Fonte: O autor

Teste n°	Problema	Resultado Heurística	Tempo (ms)	Resultado PIM	Tempo (ms)	GAP Solução	GAP Tempo
1	7x7_Ex01	76	2559	73	17250	4,11%	14,83%
2	7x7_Ex02	88	2974	85	893480	3,53%	0,33%
3	7x7_Ex03	85	3117	80	878360	6,25%	0,35%
4	7x7_Ex04	87	2671	84	4089970	3,57%	0,07%
5	7x7_Ex05	91	2506	89	486150	2,25%	0,52%
6	7x7_Ex06	88	2840	83	1610870	6,02%	0,18%
7	7x7_Ex07	86	2721	81	173110	6,17%	1,57%
8	7x7_Ex08	81	3017	76	39310	6,58%	7,67%
9	7x7_Ex09	90	2337	85	740360	5,88%	0,32%
10	7x7_Ex10	93	3185	89	7801580	4,49%	0,04%
11	7x7_Ex11	87	2395	81	174000	7,41%	1,38%
12	7x7_Ex12	86	2597	84	209140	2,38%	1,24%
13	7x7_Ex13	78	2821	76	62960	2,63%	4,48%
14	7x7_Ex14	77	2457	73	1118890	5,48%	0,22%
15	7x7_Ex15	77	2437	73	292840	5,48%	0,83%
16	7x7_Ex16	97	2960	91	2218070	6,59%	0,13%
17	7x7_Ex17	95	2676	87	252020	9,20%	1,06%
18	7x7_Ex18	90	2566	85	2349340	5,88%	0,11%
19	7x7_Ex19	85	2595	80	107560	6,25%	2,41%
20	7x7_Ex20	95	2536	91	10499620	4,40%	0,02%
21	7x7_Ex21	92	2555	90	234020	2,22%	1,09%
22	7x7_Ex22	78	2657	73	282000	6,85%	0,94%
23	7x7_Ex23	94	2686	89	1666750	5,62%	0,16%
24	7x7_Ex24	83	2695	78	229370	6,41%	1,17%
25	7x7_Ex25	91	2738	87	5711480	4,60%	0,05%
26	7x7_Ex26	87	2426	83	72740	4,82%	3,34%
27	7x7_Ex27	81	2576	74	1485180	9,46%	0,17%
28	7x7_Ex28	81	2681	78	2403050	3,85%	0,11%
29	7x7_Ex29	90	3085	85	2153940	5,88%	0,14%
30	7x7_Ex30	92	2591	88	587660	4,55%	0,44%

Tabela B.6: Resultados Testes 8x8

Fonte: O autor

Teste nº	Problema	Resultado Heurística	Tempo (ms)	Resultado PIM (Tempo 42s.)	GAP Solução
1	8x8_Ex01	115	3497	120	-4,17%
2	8x8_Ex02	103	3448	104	-0,96%
3	8x8_Ex03	109	4193	105	3,81%
4	8x8_Ex04	97	3894	99	-2,02%
5	8x8_Ex05	92	3386	96	-4,17%
6	8x8_Ex06	110	3435	108	1,85%
7	8x8_Ex07	107	3830	112	-4,46%
8	8x8_Ex08	117	3390	106	10,38%
9	8x8_Ex09	109	3406	115	-5,22%
10	8x8_Ex10	114	3890	130	-12,31%
11	8x8_Ex11	110	3368	114	-3,51%
12	8x8_Ex12	108	3957	108	0,00%
13	8x8_Ex13	108	3896	120	-10,00%
14	8x8_Ex14	111	3972	142	-21,83%
15	8x8_Ex15	109	3413	124	-12,10%
16	8x8_Ex16	107	3355	103	3,88%
17	8x8_Ex17	111	3882	113	-1,77%
18	8x8_Ex18	91	3498	92	-1,09%
19	8x8_Ex19	108	3448	106	1,89%
20	8x8_Ex20	101	3351	105	-3,81%
21	8x8_Ex21	90	3471	86	4,65%
22	8x8_Ex22	106	3976	108	-1,85%
23	8x8_Ex23	109	3433	117	-6,84%
24	8x8_Ex24	96	3397	100	-4,00%
25	8x8_Ex25	107	3367	116	-7,76%
26	8x8_Ex26	102	3442	110	-7,27%
27	8x8_Ex27	103	3898	114	-9,65%
28	8x8_Ex28	112	3942	118	-5,08%
29	8x8_Ex29	120	3415	133	-9,77%
30	8x8_Ex30	117	3337	118	-0,85%

Tabela B.7: Resultados Testes 9x9

Fonte: O autor

Teste n°	Problema	Resultado Heurística	Tempo (ms)	Resultado PIM (Tempo 49s.)	GAP Solução
1	9x9_Ex01	114	4677	158	-27,85%
2	9x9_Ex02	137	4859	176	-22,16%
3	9x9_Ex03	126	4798	179	-29,61%
4	9x9_Ex04	124	4708	-	-
5	9x9_Ex05	116	4743	157	-26,11%
6	9x9_Ex06	127	4824	172	-26,16%
7	9x9_Ex07	120	4825	126	-4,76%
8	9x9_Ex08	130	4652	203	-35,96%
9	9x9_Ex09	132	4723	166	-20,48%
10	9x9_Ex10	126	4705	171	-26,32%
11	9x9_Ex11	117	4698	182	-35,71%
12	9x9_Ex12	120	4734	260	-53,85%
13	9x9_Ex13	125	4701	139	-10,07%
14	9x9_Ex14	117	4801	139	-15,83%
15	9x9_Ex15	123	4755	152	-19,08%
16	9x9_Ex16	130	4716	141	-7,80%
17	9x9_Ex17	116	4800	128	-9,38%
18	9x9_Ex18	118	4729	-	-
19	9x9_Ex19	118	4698	169	-30,18%
20	9x9_Ex20	119	4822	152	-21,71%
21	9x9_Ex21	114	4768	160	-28,75%
22	9x9_Ex22	130	4631	-	-
23	9x9_Ex23	128	4727	176	-27,27%
24	9x9_Ex24	118	4710	238	-50,42%
25	9x9_Ex25	121	4743	141	-14,18%
26	9x9_Ex26	128	4750	184	-30,43%
27	9x9_Ex27	129	4743	-	-
28	9x9_Ex28	114	4774	150	-24,00%
29	9x9_Ex29	117	4698	-	-
30	9x9_Ex30	115	4835	-	-

Tabela B.8: Resultados Testes 10x10

Fonte: O autor

Teste nº	Problema	Resultado Heurística	Tempo (ms)	Resultado PIM 1min	% de Melhoria	Resultado PIM 2min	% de Melhoria
1	10x10_Ex01	139	6462	130	6,47%	130	6,47%
2	10x10_Ex02	143	6593	132	7,69%	130	9,09%
3	10x10_Ex03	142	6443	135	4,93%	135	4,93%
4	10x10_Ex04	140	6698	136	2,86%	136	2,86%
5	10x10_Ex05	132	6509	127	3,79%	123	6,82%
6	10x10_Ex06	143	6528	141	1,40%	141	1,40%
7	10x10_Ex07	135	6614	135	0,00%	133	1,48%
8	10x10_Ex08	129	6589	123	4,65%	123	4,65%
9	10x10_Ex09	137	6485	130	5,11%	130	5,11%
10	10x10_Ex10	133	6499	130	2,26%	128	3,76%
11	10x10_Ex11	142	6912	136	4,23%	130	8,45%
12	10x10_Ex12	138	6463	134	2,90%	133	3,62%
13	10x10_Ex13	138	6541	131	5,07%	131	5,07%
14	10x10_Ex14	133	6539	123	7,52%	122	8,27%
15	10x10_Ex15	128	6553	125	2,34%	125	2,34%
16	10x10_Ex16	136	6516	130	4,41%	130	4,41%
17	10x10_Ex17	133	6493	130	2,26%	130	2,26%
18	10x10_Ex18	137	6443	131	4,38%	131	4,38%
19	10x10_Ex19	151	6435	144	4,64%	144	4,64%
20	10x10_Ex20	152	6536	144	5,26%	144	5,26%
21	10x10_Ex21	142	6522	135	4,93%	134	5,63%
22	10x10_Ex22	143	6495	133	6,99%	133	6,99%
23	10x10_Ex23	138	6548	130	5,80%	129	6,52%
24	10x10_Ex24	145	6519	132	8,97%	132	8,97%
25	10x10_Ex25	129	6555	126	2,33%	126	2,33%
26	10x10_Ex26	130	6432	123	5,38%	123	5,38%
27	10x10_Ex27	135	6526	124	8,15%	124	8,15%
28	10x10_Ex28	133	6417	125	6,02%	119	10,53%
29	10x10_Ex29	136	6627	129	5,15%	126	7,35%
30	10x10_Ex30	135	6563	131	2,96%	129	4,44%

Tabela B.9: Resultados Testes 15x15

Fonte: O autor

Teste nº	Problema	Resultado Heurística	Tempo (ms)	Resultado PIM 1min	% de Melhoria	Resultado PIM 2min	% de Melhoria
1	15x15_Ex01	211	22589	210	0,47%	209	0,95%
2	15x15_Ex02	232	23360	232	0,00%	217	6,47%
3	15x15_Ex03	218	22607	218	0,00%	210	3,67%
4	15x15_Ex04	224	22801	224	0,00%	210	6,25%
5	15x15_Ex05	223	22607	223	0,00%	209	6,28%
6	15x15_Ex06	219	22638	219	0,00%	210	4,11%
7	15x15_Ex07	224	22658	224	0,00%	203	9,38%
8	15x15_Ex08	214	22773	214	0,00%	198	7,48%
9	15x15_Ex09	215	22684	215	0,00%	211	1,86%
10	15x15_Ex10	217	22654	216	0,46%	207	4,61%
11	15x15_Ex11	235	22769	235	0,00%	227	3,40%
12	15x15_Ex12	227	22528	227	0,00%	221	2,64%
13	15x15_Ex13	223	22616	223	0,00%	219	1,79%
14	15x15_Ex14	221	22517	218	1,36%	212	4,07%
15	15x15_Ex15	218	22515	216	0,92%	198	9,17%
16	15x15_Ex16	217	22495	217	0,00%	206	5,07%
17	15x15_Ex17	223	22585	223	0,00%	214	4,04%
18	15x15_Ex18	217	22550	217	0,00%	211	2,76%
19	15x15_Ex19	216	22614	213	1,39%	210	2,78%
20	15x15_Ex20	222	22573	219	1,35%	212	4,50%
21	15x15_Ex21	218	22320	218	0,00%	207	5,05%
22	15x15_Ex22	231	22733	231	0,00%	224	3,03%
23	15x15_Ex23	215	22572	215	0,00%	202	6,05%
24	15x15_Ex24	220	22918	220	0,00%	215	2,27%
25	15x15_Ex25	215	22662	215	0,00%	211	1,86%
26	15x15_Ex26	208	22775	208	0,00%	205	1,44%
27	15x15_Ex27	226	22834	226	0,00%	215	4,87%
28	15x15_Ex28	214	22708	213	0,47%	201	6,07%
29	15x15_Ex29	225	22565	225	0,00%	213	5,33%
30	15x15_Ex30	231	23244	229	0,87%	216	6,49%

Tabela B.10: Resultados Testes 20x20

Fonte: O autor

Teste nº	Problema	Resultado Heurística	Tempo (ms)	Resultado PIM 1min	% de Melhoria	Resultado PIM 2min	% de Melhoria
1	20x20_Ex01	282	58250	282	0,00%	281	0,35%
2	20x20_Ex02	296	58102	296	0,00%	296	0,00%
3	20x20_Ex03	312	58591	312	0,00%	312	0,00%
4	20x20_Ex04	315	58853	315	0,00%	313	0,63%
5	20x20_Ex05	305	58347	305	0,00%	305	0,00%
6	20x20_Ex06	328	58005	328	0,00%	328	0,00%
7	20x20_Ex07	303	58498	303	0,00%	297	1,98%
8	20x20_Ex08	300	58376	300	0,00%	300	0,00%
9	20x20_Ex09	300	57776	300	0,00%	297	1,00%
10	20x20_Ex10	302	57938	302	0,00%	301	0,33%
11	20x20_Ex11	318	59671	318	0,00%	307	3,46%
12	20x20_Ex12	310	57978	308	0,65%	299	3,55%
13	20x20_Ex13	323	58387	323	0,00%	321	0,62%
14	20x20_Ex14	326	58037	326	0,00%	314	3,68%
15	20x20_Ex15	299	58951	299	0,00%	296	1,00%
16	20x20_Ex16	315	58043	315	0,00%	308	2,22%
17	20x20_Ex17	324	57964	324	0,00%	322	0,62%
18	20x20_Ex18	304	58324	304	0,00%	304	0,00%
19	20x20_Ex19	315	58179	315	0,00%	305	3,17%
20	20x20_Ex20	308	58182	308	0,00%	301	2,27%
21	20x20_Ex21	315	59693	315	0,00%	310	1,59%
22	20x20_Ex22	316	59477	316	0,00%	316	0,00%
23	20x20_Ex23	307	58475	307	0,00%	307	0,00%
24	20x20_Ex24	322	60473	322	0,00%	320	0,62%
25	20x20_Ex25	309	58667	309	0,00%	301	2,59%
26	20x20_Ex26	299	58034	299	0,00%	294	1,67%
27	20x20_Ex27	305	59377	305	0,00%	305	0,00%
28	20x20_Ex28	299	58124	299	0,00%	290	3,01%
29	20x20_Ex29	311	59158	311	0,00%	311	0,00%
30	20x20_Ex30	314	59339	314	0,00%	303	3,50%